UNIVERSIDAD DE CARABOBO
FACULTAD DE CIENCIAS Y TECNOLOGÍA
DEPARTAMENTO DE QUÍMICA

Productos de programación en lenguaje C relacionados con QTAIM

PRODUCCIÓN INTELECTUAL PRESENTADA COMO CREDENCIAL DE MÉRITO
PARA ASCENDER A LA CATEGORÍA DE PROFESOR ASOCIADO

David Vega M.

C.I. 15.529.951

Bárbula, Abril de 2014

ÍNDICE

Introducción 1

Artículo 1: Biblioteca en lenguaje C para el estudio topológico de la densidad de carga electrónica 3

Artículo 2: AIM-UC: Una aplicación para el análisis QTAIM 23

Apéndices (artículos originales) 35

INTRODUCCIÓN

Este trabajo de ascenso consiste en dos artículos de investigación, publicados en revistas internacionales, con productos de programación relacionados con la teoría cuántica de átomos en moléculas (QTAIM por sus siglas en inglés). El primer artículo, "C Library for Topological Study of the Electronic Charge Density" (*Journal of Computational Chemistry* **2012**, *33*, 2526–2531), se refiere a una biblioteca escrita en lenguaje C para uso de programadores interesados en hacer aplicaciones para estudios topológicos. El segundo artículo "AIM-UC: An application for QTAIM analysis" (*Journal of Computational Methods in Sciences and Engineering* **2014**, *14* 131–136) se trata de la presentación de una aplicación para realizar gráficos relacionados con QTAIM. Ambos productos, pueden obtenerse en el sitio web:

http://alfa.facyt.uc.edu.ve/quimicomp/

Artículo 1

D. Vega, Y. Aray, J. Rodríguez “C Library for Topological Study of the Electronic Charge Density” *Journal of Computational Chemistry* **2012**, *33*, 2526–2531.

DOI: 10.1002/jcc.23083

http://onlinelibrary.wiley.com/doi/10.1002/jcc.23083/abstract

Biblioteca en lenguaje C para el estudio topológico de la densidad de carga electrónica

David Vega, Yosslen Aray, y Jesús Rodríguez

El estudio topológico de la densidad de carga electrónica es útil para obtener información sobre los tipos de enlaces (iónicos o covalentes) y las cargas atómicas en una molécula o cristal. Para este estudio es necesario calcular, en cada punto del espacio, la densidad electrónica y sus derivadas hasta de segundo orden. En el presente trabajo se describe una metodología para estos cálculos basada en retículas 3D. La biblioteca se basa en una interpolación de Lagrange multidimensional en una rejilla regular, implementada para 3 dimensiones. Al diferenciar el polinomio resultante, se obtienen las fórmulas del vector gradiente, la matriz Hessiana y la Laplaciana para cada punto del espacio. Se programaron funciones más complejas tales como el método de Newton–Raphson para encontrar los puntos críticos (aquellos puntos donde el gradiente es nulo) y el método de Cash–Karp Runge–Kutta utilizado para construir los caminos de gradiente. Dado que en algunos cristales la celda unidad tiene ángulos diferentes de 90 °, la biblioteca descrita incluye transformaciones lineales para corregir el gradiente y de Hessiana cuando la red está distorsionada (inclinada). Las funciones también se han desarrollado para manejar archivos que contienen retículas (grd del programa DMol®, CUBE del programa Gaussian® y CHGCAR del programa VASP®). Cada uno de estos archivos contiene los datos de una propiedad electrónica molecular o de cristal (tales como densidad de carga, densidad de espín, potencial electrostático y otros) en una rejilla tridimensional. La biblioteca puede ser adaptada para hacer el estudio topológico en cualquier rejilla tridimensional regular mediante la modificación del código de estas funciones.

Introducción

La “teoría cuántica de átomos en moléculas” (QTAIM) de Bader et al. es muy útil para obtener la información química a partir de la densidad de carga [1-8]. QTAIM es una teoría firme, rigurosa y mecánico cuántica bien definida basada en observables, tales como la densidad de electrones o campos de densidad de energía. La mayoría de las teorías modernas de enlaces se basan, de una manera u otra, en la partición de la carga (o densidad electrónica) entre los diferentes centros nucleares en estudio, por lo general por medio de Mulliken - es decir, la densidad proyectada de estados en un análisis de sólidos. De esta manera, una cantidad importante de los modelos interpretativos de comportamiento químico se basa en conceptos que se sabe que están muy mal definidos, y para dar respuestas extremadamente dependientes de toda una jerarquía de aproximaciones [6]. QTAIM proporciona un vínculo cuantitativo entre la densidad total de electrones (independientemente de cómo se generó: calculada o experimental) y las propiedades físicas importantes de una molécula, sin pasar por la función de onda en el análisis. QTAIM es una metodología libre del concepto de orbital. En particular, proporciona una definición rigurosa de enlace químico y estructura geométrica para todos los tipos de moléculas y sólidos, y ha demostrado ser útil en el análisis de las propiedades físicas de los aislantes, metales puros y aleaciones [4-6]. Densidades experimentales de alta calidad de los minerales [9, 10], covalentes [11], metálicos [12], y cristales moleculares [13, 14] han sido analizadas en términos de conceptos de QTAIM. Además, también se ha reportado cálculos QTAIM sobre los metales simples [15], aleaciones, y fases intermetálicas [16, 17], así como en cristales moleculares [18, 19], covalentes e iónicos [6, 11]. Se ha publicado sobre paquetes de programas que calculan y dibujan caminos de gradiente, isolíneas de la densidad y su laplaciana [20-28], utilizando métodos computacionales similares. Sin embargo, solo pocos de ellos tiene código fuente abierto (Aimpac [25], Multiwfn [26], DGrid [27], Critic [28]).

El estudio topológico consiste en determinar y caracterizar los puntos críticos, caminos de enlace, superficies de flujo cero, mapas de gradiente y cuencas atómicas. Los puntos críticos son aquellos donde el gradiente es cero (entre estos puntos hay mínimos y máximos locales,

y puntos de ensilladura) y son fundamentales para el estudio topológico. Los máximos locales corresponden a las posiciones de núcleos atómicos. Los puntos de ensilladura, mínimos en una dirección y máximos en el plano perpendicular se denominan puntos críticos de enlace, los máximos en una dirección y mínimos en el plano perpendicular se denominan puntos críticos de anillo, y los mínimos locales se denominan puntos críticos de caja. Una representación de enlaces y superficies interatómicas se obtienen dibujando los caminos de gradientes cercanos al punto crítico de enlace. Para encontrar esos puntos es necesario obtener el vector gradiente, **g**(**r**) y la matriz Hessiana, **H**(**r**) en cualquier punto **r** del espacio (usando el método de Newton–Raphson, descrito posteriormente).

$$\mathbf{r}=\begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \mathbf{g}(\mathbf{r})=\nabla f(\mathbf{r})=\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} \qquad \mathbf{H}(\mathbf{r})=\begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial x \partial z} & \frac{\partial^2 f}{\partial y \partial z} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix} \qquad (1)$$

Con el propósito de reemplazar rutinas de cálculo de gradiente y Hessiana en paquetes de programas que calculan puntos críticos y sus propiedades, la biblioteca comenzó a construirse a principios del 2000 usando Fortran 77. Para el desarrollo integral de nuestros propios programas, posteriormente, la biblioteca fue traducida al lenguaje C. El contenido de esta biblioteca en C fue antes reportada [29], no en detalle, pero incluyendo algunos ejemplos de cálculo. También ha sido usada como herramienta en estudios por nuestro grupo[30]. En este trabajo se describe de forma exhaustiva la biblioteca C, proporcionando el código fuente, modo de empleo y algunas funciones adicionales para la lectura de las retículas en distintos formatos.

La biblioteca contiene funciones para calcular el vector gradiente y la matriz Hessiana, y otras funciones útiles para el estudio topológico.

Aproximación a la función y derivadas

Comenzando con la fórmula de aproximación polinomial de Lagrange [31, 32], ec. (2):

$$P(x)=\sum_{k=1}^{n}\frac{\prod_{j=1,j\neq k}^{n}(x-x_j)}{\prod_{j=1,j\neq k}^{n}(x_k-x_j)}f_k \qquad (2)$$

Donde n es el número de puntos (x_j, f_j) usados en la interpolación y $P(x)$ es un polinomio de grado $n - 1$ que pasa por los n puntos. Si los valores x_j están igualmente espaciados [29] (Fig. 1), entonces $x_j = x_0 + j\,h$, donde h es la distancia entre los puntos.

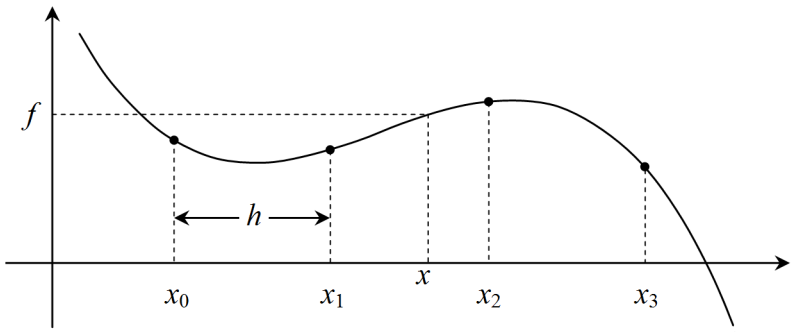

Figura 1. El polinomio de Lagrange pasa por los puntos del arreglo (puntos negros). Cuando el polinomio es evaluado en otro punto (x), se obtiene un valor aproximado de la función (f). Cuando es posible, es conveniente escoger el arreglo de puntos de modo que el valor de x esté en el intervalo central [29]. En este caso, hay dos puntos con valor x_i menor que x y dos con valor mayor.

Definiendo $s = (x - x_\alpha)/h$, con un índice constante α (es conveniente escoger α de modo que x_α and $x_{\alpha+1}$ sean los puntos centrales del arreglo: n es par y $\alpha = n/2 - 1$), resolviendo para x ($x = x_\alpha + s \cdot h$) y sustituyendo en el polinomio de Lagrange, Ec. (2):

$$P=\sum_{k=0}^{n-1} w_{k,n}(s)\, f_k \quad (3)$$

donde $$w_{k,n}(s)=\frac{\prod_{j=0,j\neq k}^{n-1}(\alpha-j+s)}{\prod_{j=0,j\neq k}^{n-1}(k-j)} \quad (4)$$

$w_{k,n}(s)$ es un polinomio de grado $n-1$ en s.

Para obtener las aproximaciones de las derivadas de la función respecto a x, $w_{k,n}(s)$ se deriva respecto a s, como se muestra:

$$\frac{dP}{dx}=\frac{dP}{ds}\frac{ds}{dx}=\frac{1}{h}\frac{dP}{ds}=\frac{1}{h}\sum_{k=0}^{n-1}\frac{dw_{k,n}(s)}{ds} f_k \quad (5)$$

Para derivadas de mayor orden,

$$\frac{d^v P}{dx^v}=\frac{1}{h^v}\sum_{k=0}^{n-1} w_{k,n}^{(v)}(s)\, f_k \quad (6)$$

donde

$$w_{k,n}^{(v)}(s)=\frac{d^v w_{k,n}(s)}{ds^v}$$

En la interpolación y la aproximación a las derivadas (en la biblioteca desarrollada), los polinomios $w_{k,n}^{(v)}(s)$ se calcularon para cuatro puntos ($n = 4$). Ver tabla 1.

Expresando los polinomios de Lagrange como una suma de productos de f_k y el peso $w_{k,n}(s)$, ec. (2), permite obtener las derivadas en una forma directa, ec (5). También, calculando el producto tensorial [31] de los pesos $w_{k,n}(s)$, puede obtenerse una aproximación para un arreglo regular multidimencional. Para una retícula tridimensional (3D), se calcula para cada

dimensión el valor de s_i ($s_1 = (x - x_\alpha)/h_1$, $s_2 = (y - y_\alpha)/h_2$, $s_3 = (z - z_\alpha)/h_3$), entonces se evalúa el polinomio $w_{k,n}(s)$. Estos polinomios multiplican los valores de la función en cada punto de la retícula ($f_{k_1k_2k_3}$):

$$P = \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} \sum_{k_3=0}^{n_3-1} w_{k_1,n_1}(s_1) \cdot w_{k_2,n_2}(s_2) \cdot w_{k_3,n_3}(s_3) \cdot f_{k_1k_2k_3} \qquad (7)$$

donde n_1, n_2 y n_3 son los números de puntos usados en la interpolación para cada dimensión. Derivando el polinomio resultante respecto a s_i, se obtiene la aproximación a cualquiera de las derivadas respecto a las variables x, y, z.

$$\frac{\partial P^{v_1+v_2+v_3}}{\partial x^{v_1} \partial y^{v_2} \partial z^{v_3}} = \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} \sum_{k_3=0}^{n_3-1} w_{k_1,n_1}^{(v_1)}(s_1) \cdot w_{k_2,n_2}^{(v_2)}(s_2) \cdot w_{k_3,n_3}^{(v_3)}(s_3) \cdot f_{k_1k_2k_3} \qquad (8)$$

La ec. (8) es equivalente a la ec. (7), cuando $v_1 = v_2 = v_3 = 0$, entendiendo que $w_{k,n}^{(0)}(s) = w_{k,n}(s)$.

La biblioteca desarrollada consiste principalmente en rutinas que, dados los valores de la función en una retícula regular 3D, calcula el valor aproximado de la función, vector gradiente, y matriz Hessiana, en cualquier punto del espacio dentro de los límites de la retícula. Estas rutinas son implantaciones de la ec. (8) con $n_1 = n_2 = n_3 = 4$ (interpolación tricúbica de Lagrange). A diferencia de la interpolación trilineal [33], esta permite aproximaciones para derivadas de segundo orden (ver tabla 1).

TABLA 1. Polinomios $w_{k,n}^{(v)}(s)$, definidos en ec. (7) y ec. (4), con $n = 4$ y $\alpha = 1$, y $w_{k,n}^{(0)}(s) = w_{k,n}(s)$.

	$k = 0$	$k = 1$	$k = 2$	$k = 3$
$v = 0$	$(-s^3 + 3s^2 - 2s)/6$	$(s^3 - 2s^2 - s + 2)/2$	$(-s^3 + s^2 + 2s)/2$	$(s^3 - s)/6$
$v = 1$	$(-3s^2 + 6s - 2)/6$	$(3s^2 - 4s - 1)/2$	$(-3s^2 + 2s + 2)/2$	$(3s^2 - 1)/6$
$v = 2$	$-s + 1$	$3s - 2$	$-3s + 1$	s

Retícula Inclinada

El término “regular” para la retícula, significa que el espacio entre los puntos (h_i) es constante a lo largo de cualquier dimensión particular, sin embargo, puede ser diferente al de otra dimensión. Además, la retícula puede estar inclinada (fig. 2). Esto significa que, al menos uno de los ángulos entre los ejes de la dimensión es diferente de 90 ° (esto es usual para la celda unidad de muchos sólidos cristalinos). En este caso, siempre tomamos un eje paralelo al eje *x* y otro paralelo al plano *xy*.

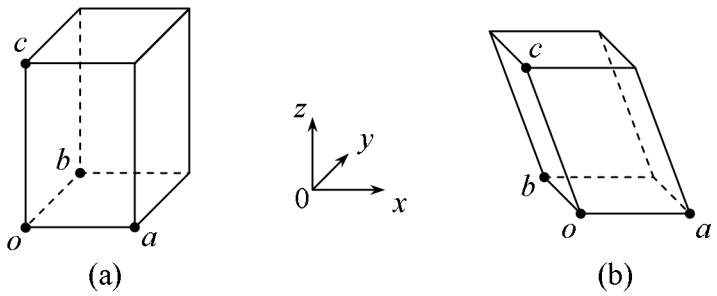

Figura 2. a) Retícula no inlinada, *o* está localizado en el origen, *a* en el eje *x*, *b* en el eje *y*, y *c* en el eje *z*. b) Retícula inclinada, *o* está localizado en el origen, *a* en el eje *x*, *b* en el plano *xy*, y *c* puede estar en cualquier sitio.

Para calcular el gradiente evaluado en el punto r del espacio en una retícula inclinada, es necesario realizar dos transformaciones lineales [34], primero, debe calcularse las coordenadas del punto en el sistema de coordenadas inclinadas (**r'**), $\mathbf{Ar} = \mathbf{r'}$, donde A es la matriz que transforma las coordenadas ordinarias (no inclinadas) a coordenadas inclinadas. Entonces, se determina el vector gradiente (**g'**) calculando las derivadas, usando la ec. (8). Finalmente, el gradiente es transformado de vuelta al sistema de coordenadas ordinarias, multiplicando por la transpuesta de **A**, $\mathbf{g} = \mathbf{A}^{\mathrm{T}}\mathbf{g'}$.

Para calcular la matriz Hessiana (**H**), es también necesario transformar el punto **r** al sistema

de coordenadas inclinadas. De aquí se calcula **H'** (usando la ec. (8) para obtener las derivadas de segundo orden), **H'** es finalmente transformado de vuelta al sistema de coordenadas ordinarias: $\mathbf{H} = \mathbf{A}^T\mathbf{H}'\mathbf{A}$.

Estas ecuaciones son bastante generales, cuando existe una transformación lineal de un sistema coordinado a otro, sin embargo, en la biblioteca desarrollada, el único caso considerado es cuando el sistema de coordenadas inclinadas satisface las condiciones dadas (fig. 2). En este caso , la matriz de transformación es siempre una matriz triangular superior.

Aproximación al logaritmo de la función

La densidad de carga electrónica tiene características particulares, por ejemplo, su valor nunca es negativo, y tiene un comportamiento exponencial cerca del núcleo atómico. El gran incremento del valor de la densidad causa que el polinomio interpolado oscile produciendo valores negativos en la vecindad del núcleo (ver fig. 3b). Cambiando el valor de la retícula ($f_{k_1k_2k_3}$) por su logaritmo en la interpolación previamente descrita, ec. (7), y finalmente tomando el antilogaritmo del valor interpolado (P), es posible evitar la condición de oscilación.

En el caso de derivadas, se usan las ec. (9) y (10).

$$\frac{\partial f}{\partial u} = \frac{\partial p}{\partial u} e^p \qquad (9)$$

$$\frac{\partial^2 f}{\partial u \partial v} = \left(\frac{\partial^2 p}{\partial u \partial v} + \frac{\partial p}{\partial u}\frac{\partial p}{\partial v} \right) e^p \qquad (10)$$

donde u y v son x, y o z, y p es la interpolación del logaritmo. Los valores $\partial p/\partial u$, $\partial p/\partial v$, y $\partial^2 p/\partial u \partial v$ son las aproximaciones a las derivadas de primer y segundo orden obtenidas por el método cuando se usan los logaritmos de los valores de la retícula.

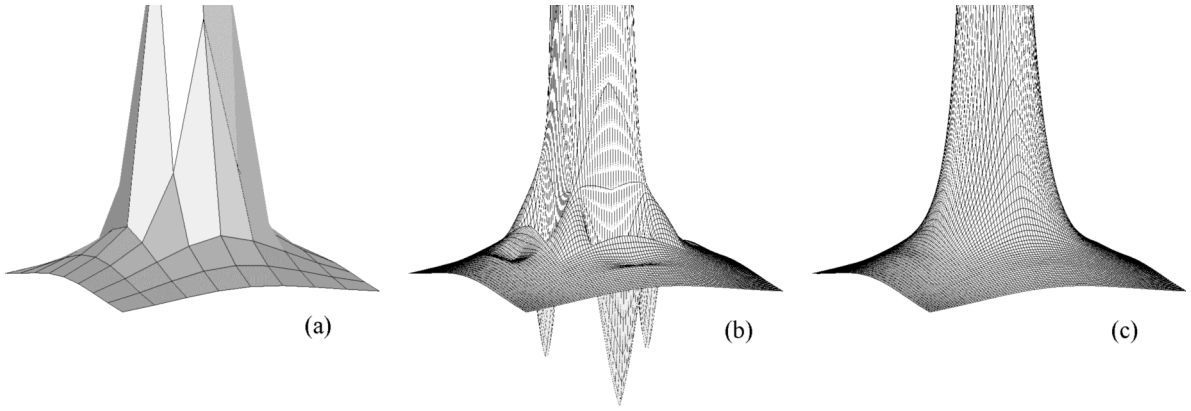

Figura 3. Gráficos de la densidad de carga electrónica en un plano que contiene al átomo de azufre de la molécula dibenzotiofeno. El plano contiene 9 × 9 puntos de la retícula, la distancia entre los puntos es 5×10^{-12} m. a) Sin interpolación, b) Usando interpolación polinomial. c) Usando interpolación de logaritmos.

Métodos que requieren al vector gradiente y la matriz Hessiana

Los métodos siguientes se utilizan en el estudio topológico y requieren al vector gradiente y la matriz Hessiana.

Método de Newton–Raphson

Debido al hecho que el gradiente de la función evaluado en un punto crítico es el vector nulo (**0**), la forma de calcular las coordenadas del punto es resolver la ecuación $\nabla\rho(\mathbf{r}_{crit}) = \mathbf{0}$. Una forma alternativa para resolver esta ecuación es usar el método de Newton–Raphson [33]. La función escalar multidimensional $\nabla\rho(\mathbf{r})$ evaluada en el punto $\mathbf{r} = \mathbf{r}_i + \mathbf{h}$ se expande en serie de Taylor:

$$\nabla\rho(\mathbf{r}) = \nabla\rho(\mathbf{r}_i) + \mathbf{H}_i\,\mathbf{h} + \text{términos de orden superior} \qquad (11)$$

donde $\mathbf{H}_i$ es la matriz Hessiana $\mathbf{H}(\mathbf{r})$ (la Jacobiana de $\nabla\rho(\mathbf{r})$) evaluada en $\mathbf{r}_i$.

Despreciando los términos de orden superior en la serie de Taylor, ec. (11), haciendo $\nabla\rho(\mathbf{r}) = \mathbf{0}$ y resolviendo para **h**:

$$\mathbf{h} = -\mathbf{H}_i^{-1}\nabla\rho(\mathbf{r}_i) \qquad (12)$$

obtenemos el vector desplazamiento (**h**).

Si la función $\rho(\mathbf{r})$ es cuadrática entonces el vector **h** comienza en el punto $\mathbf{r}_i$ y termina en el punto crítico. En general, la función no es cuadrática, así que un nuevo punto $\mathbf{r}_{i+1}$ se calcula usando:

$$\mathbf{r}_{i+1} = \mathbf{r}_i + t\,\mathbf{h} \qquad (13)$$

donde t es un valor pequeño, menor que 1.

El calculo de acuerdo a las ecuaciones (10) y (11) se itera hasta que $\nabla\rho(\mathbf{r}_i)$ es igual al vector **0** o tiene un módulo muy pequeño. Entonces, $\mathbf{r}_i$ es un punto crítico o uno muy cercano.

En el algoritmo desarrollado, el módulo de **h** tiene una cota superior. Cuando $|\mathbf{h}|$ es mayor que la cota, se calcula t de modo que $|t\,\mathbf{h}|$ es igual a la cota (en otro caso $t = 1$). Para cada iteración, la cota se disminuye por medio de una progresión geométrica, en la cual la razón debe ser menor que 1. Haciendo esto, se evitan grandes oscilaciones cerca de los puntos críticos, cuando el gradiente tiene un valor de módulo elevado en las vecindades del punto crítico. El número máximo de iteraciones y la longitud de la trayectoria se definen por el usuario.

Método de Cash–Karp Runge–Kutta de quinto orden

Los caminos de gradiente para la construcción del grafo molecular y las superficies

interatómicas son soluciones de la ec. diferencial (14) [1].

$$d\mathbf{r}(s)/ds = \nabla\rho(\mathbf{r}(s)) \quad (14)$$

La solución de la ec. (14) es una curva paramétrica en R^3, la cual es única para un dado valor inicial **r**. Una solución numérica se obtiene con el método de Cash–Karp Runge–Kutta de quinto orden [33]. La forma general de este método es

$$\mathbf{r}_{n+1} = \mathbf{r}_n + c_1\,\mathbf{k}_1 + c_2\,\mathbf{k}_2 + c_3\,\mathbf{k}_3 + c_4\,\mathbf{k}_4 + c_5\,\mathbf{k}_5 + c_6\,\mathbf{k}_6 \quad (15)$$

donde $\mathbf{r}_n = (x_n, y_n, z_n)$ y los valores $\mathbf{k}_j$ con un pequeño desplazamiento h son:

$$\mathbf{k_1} = h\nabla\rho(\mathbf{r})|_{\mathbf{r}=\mathbf{r_n}}$$
$$\mathbf{k_2} = h\nabla\rho(\mathbf{r})|_{\mathbf{r}=\mathbf{r_n}+b_{21}\mathbf{k_1}}$$
$$\cdots$$
$$\mathbf{k_6} = h\nabla\rho(\mathbf{r})|_{\mathbf{r}=\mathbf{r_n}+b_{61}\mathbf{k_1}+\cdots+b_{65}\mathbf{k_5}}$$

Los valores particulares de varias constantes (c_j, b_{ij}) pueden encontrarse en “Numerical recipes” [33]. Escogiendo el desplazamiento h como un valor pequeño adecuado, el conjunto de puntos $\mathbf{r}_n$ se ajusta lo suficientemente cercano al camino de gradiente.

Discontinuidades en derivadas

Las discontinuidades en derivadas de los polinomios de Lagrange por intervalos, no permite encontrar algunos puntos críticos localizados en las caras de los cubos de la retícula. Por el contrario, la interpolación con “splines” asegura la continuidad de derivadas, pero requiere un arreglo para almacenar las derivadas segundas y resolver varios sistemas de ecuaciones que involucran todos los datos de la retícula [33]. Para ahorrar tiempo y almacenamiento en memoria, preferimos usar la interpolación de Lagrange, en vez del método de “splines”, y resolver el problema de la discontinuidad como explicaremos adelante.

Se toman siempre cuatro puntos a lo largo de cada dimensión (4 × 4 × 4 para una retícula 3D), de modo que el punto, en el cual será calculada la derivada, en la medida de lo posible, debe estar dentro del intervalo central (fig. 4). Esto causa un cambio del polinomio de interpolación cuando se deja el intervalo central. En general, esta discontinuidad no representa una desventaja seria, no obstante, el método de Newton–Raphson puede fallar cuando el punto crítico está exactamente en los planos límites de los intervalos (con un valor de s_i igual a cero, ec. (7) y (8)), en cuyo caso es posible que el módulo del gradiente sea siempre superior al criterio de convergencia seleccionado.

En la biblioteca desarrollada, existe una opción que permite continuar con el polinomio de interpolación previamente usado, cuando el punto se mueve afuera (no más del 10%) de la longitud del intervalo en cada dimensión. Esto permite encontrar los puntos críticos que descansan en los planos límites de los intervalos (de hecho, encuentra un punto bastante cercano), sin sacrificar el criterio de convergencia que afecta la posición de los otros puntos críticos.

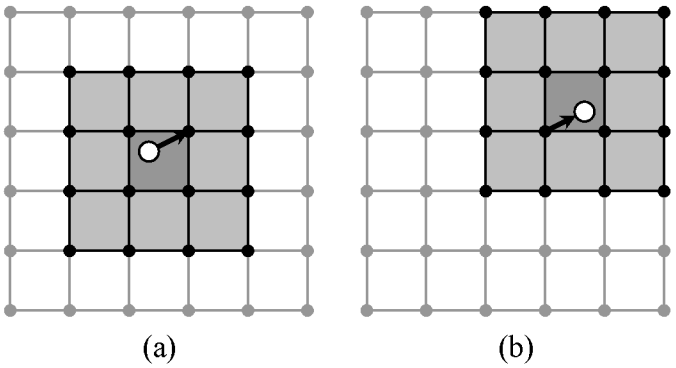

Figura 4. Cambio de polinomio de interpolación 2D. En cualquier punto dentro de la zona gris oscura, los valores $f k_1 k_2 k_3$ utilizados por el polinomio de interpolación, ec. (8), corresponden a los 16 puntos negros en la rejilla. Al calcular la pendiente o la matriz hessiana en varios puntos, comenzando en el círculo abierto en (a) y terminando en el círculo abierto en (b), el polinomio de interpolación cambia abruptamente cuando el punto alcanza la zona gris oscura en (b).

Descripción de la biblioteca

La retícula se almacena en una estructura (de lenguaje C), denominada _GRD. Están disponibles funciones que leen los archivos de datos de retícula y regresan un apuntador a la estructura _GRD con toda la información de la retícula (el espacio necesario de memoria se asigna dinámicamente en las funciones). El archivo lagrange3D4grd.c contiene la función read_grd que lee archivos grd de Dmol3 [35], una descripción más detallada de esta función y de la estructura _GRD están en el archivo lagrange3D4grd.h. El archivo GaussianCube.c contiene funciones para leer datos de retículas en formato CUBE desde archivos de salida de Gaussian [37] y Gamess [38], y el archivo CHGCARfile.c contiene la función para leer datos de retículas de archivos de salida de Vasp [39].

El argumento de la función en C es una cadena de caracteres (char * filename) que debe contener el nombre de un archivo *.grd.

Cuando la variable global _grd_latency tiene un valor diferente de cero, previene el cambio del polinomio de interpolación en los extremos del intervalo, de acuerdo a la explicado en la sección Discontinuidades en derivadas

Las funciones para calcular el valor interpolado, el gradiente y la matriz Hessiana en cualquier punto del espacio se describen en el archivo lagrange3D4grd.h. También las funciones para encontrar puntos críticos basada el el método de Newton–Raphson (descrita en la sección the Método de Newton–Raphson) y para calcular los puntos del camino de gradiente de acuerdo al método de Cash–Karp Runge–Kutta de quinto orden (descrito en la sección Método de Cash–Karp Runge–Kutta de quinto orden).

Los archivos de código de la biblioteca pueden obtenerse del sitio Web:
http://alfa.facyt.uc.edu.ve/quimicomp/

Algunos ejemplos

Los ejemplos mostrados en la fig. 5 son capturas de ventanas de algunos programas que utilizan la biblioteca descrita. Los programas fueron construidos para Windows XP con el compilador C de MinGW (MinGW es una colección de herramientas de programación disponibles gratuitamente, archivos de encabezados específicos y bibliotecas de importación que permiten producir programas nativos de Windows®, sitio Web: http://www.mingw.org/) y el ambiente de desarrollo Dev-C++ (Dev-C++ es un ambiente de desarrollo para el lenguaje de programación C/C++, para Windows, sitio Web del ambiente gratuito que incluye una versión de MinGW: http://www.bloodshed.net/devcpp.html), usando las bibliotecas OpenGL, Glut y Glui (las bibliotecas Glut y Glui para ser instaladas en Dev-C++ pueden obtenerse del sitio Web: http://www.nigels.com/glt/devpak/).

Como un ejemplo de interpolación, la figura 5a muestra el gráfico de la densidad de carga electrónica en el plano que contiene todos los átomos de la *p*-nitroanilina. El cálculo fue llevado a cabo usando el programa Gamess-US [37] al nivel HF/6-31G(2p,2d) y el espaciamiento de la retícula generada fue 0.1 bohr. Como un ejemplo de cálculo de derivadas con interpolación de logaritmos, la figura 5b es el gráfico de la Laplaciana de la densidad de carga electrónica del MoS_2 cristalino en un plano que contiene un átomo de Mo y dos de S, calculado al nivel DFT-PBE/DNP usando el programa $DMol^3$ [35]. La figura 5c es an ejemplo de cálculo de puntos críticos y caminos de gradiente para la tetraciclina (métodos de Newton–Raphson, y Cash–Karp Runge–Kutta, respectivamente) calculado usando el programa Gamess-US [37] al nivel HF/6-31G(2p,2d); el espaciamiento de la retícula fue 0.05 bohr. En la figura 5d, se muestran algunos caminos de enlace y puntos críticos del cristal MoS_2 (celda unidad inclinada).

Para la tetraciclina, se determinaron 65 puntos críticos (distintos a las posiciones nucleares) con un programa que usa esta biblioteca y también con un método analítico de densidad electrónica (Multiwfn [26]). La distancia máxima, entre las posiciones de los puntos críticos calculados con este método y con el programa Multiwfn, fue 0.0013 bohr, equivalente a 2.6 % del espaciamiento de la retícula. Y 58 de los 65 puntos tienen una distancia menor que

0.0005 bohr. Estas diferencias pueden disminuirse refinando la retícula.

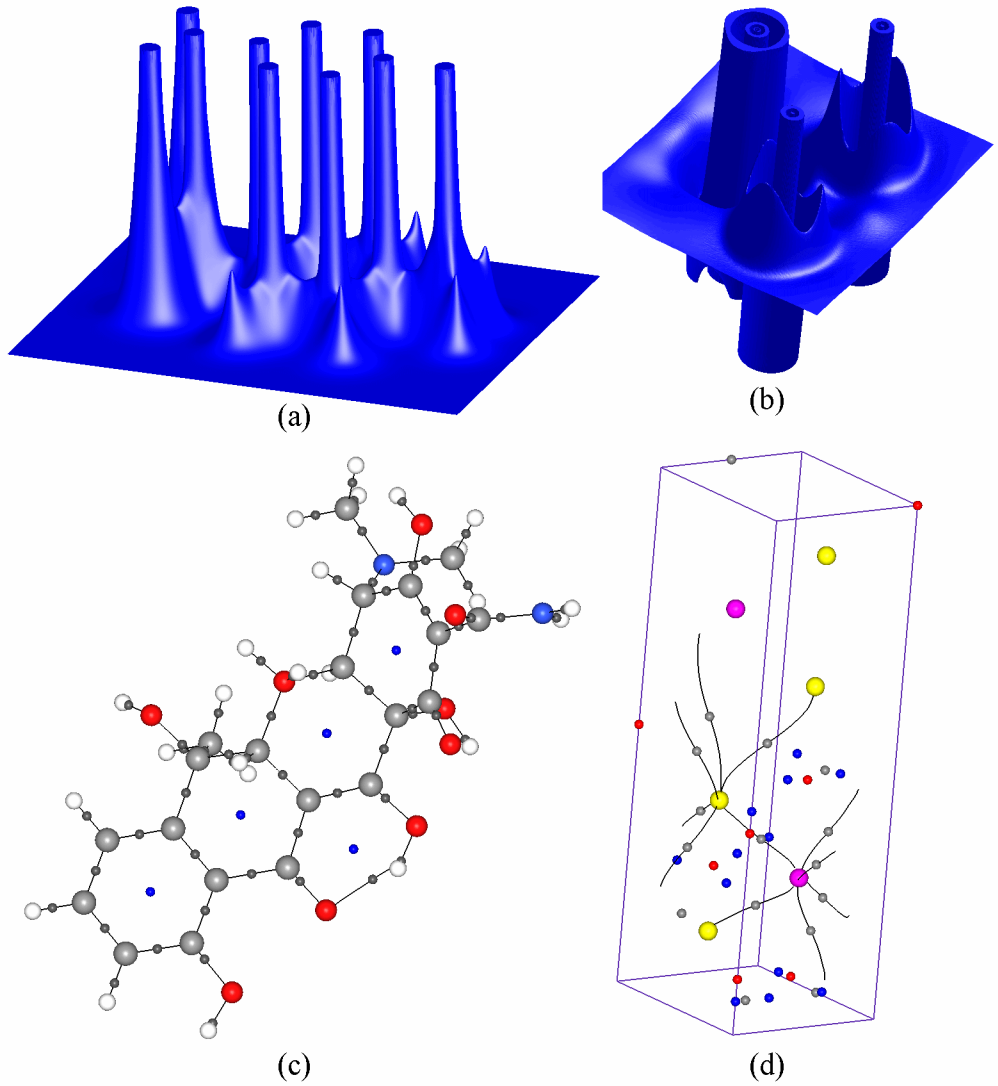

Figura 5. Ejemplos de aplicaciones de la biblioteca: a) Imagen de la densidad de carga electrónica en el plano que contiene todos los átomos de una molécula *p*-nitroanilina. b) Imagen de la Laplaciana negativa de la densidad de carga electrónica en el plano que contiene un átomo de molibdeno y dos átomos de azufre en el cristal MoS_2. c) Caminos de enlace (líneas negras), y puntos críticos (esferas grises pequeñas: puntos críticos de enlace; azules: puntos críticos de anillo) de la molécula tetraciclina. d) Algunos caminos de enlace y puntos críticos del cristal MoS_2.

Agradecimientos

Los autores expresan su agradecimiento al Profesor Oscar Valbuena por su ayuda en la preparación del manuscrito.

Referencias

[1] (a) R. F. W. Bader, Atoms in molecules-a quantum theory, Clarendon Press: Oxford, U.K., **1990**; (b) R. F. W. Bader, *Chem. Rev.* **1991**, *91*, 893.

[2] R. F. W. Bader, *J. Phys. Chem. A* **1998**, *102*, 7314.

[3] R. F. W. Bader, P. L. A. Popelier, T. A. Keith, *Angew. Chem. Int. Ed. Engl*. **1994**, *33*, 620.

[4] P. F. Zou, R. F. W. Bader, *Acta Crystallogr.* **1994**, *A50*, 714.

[5] M. E. Eberhart, *Can. J. Chem. Soc.* **1996**, *74*, 1229.

[6] M. A. Pendás, A. Costales, V. Luaña, *Phys. Rev. B* **1997**, *55*, 4275.

[7] P. L. A. Popelier, Atoms in Molecules – An Introduction; Prentice Hall: Harlow-England, **2000**.

[8] R. J. Gillespie, P. L. A. Popelier, Chemical Bonding and Molecular Geometry: From Lewis to Electrón Densities; Oxford University Press: New York, Oxford, **2001**.

[9] Y. V. Ivanov, E. L. Belokoneva, J. Protas, N. K. Hansen, V. G. Tirelson, *Acta Crystallogr. B*, **1998**, *54*, 774.

[10] T. S. Koritsanszky, P. Coppens, *Chem. Rev.* **2001**, *101*, 1583.

[11] A. Fischer, D. Tiana, W. Scherer, K. Batke, G. Eickerling, H. Svendsen, N. Bindzus, B. B. Iversen, *J. Phys. Chem. A*, **2011**, *115*, 13061-13071

[12] L. J. Farrugia, P. Macchi. *Struct. Bond* **2012**, 1, 146, 127.

[13] R. Boese, N. Niederprüm, D. Bläser, A. Maulitz, M. Y. Antipin, P. R. Mallinson, *J. Phys. Chem. B* **1997**, *101*, 5794.

[14] P. Munshi, T. N. Guru Row, *Crystallogr. Rev.* **2005**, *11*, 199.

[15] C. J. Mei, K. E. Edgecombe, V. H. Smith, A. Heilingbrunner, *Int. J. Quantum Chem.* **1993**, *48*, 287.

[16] G. H. Grosch, K. J. Range, *J. Alloys Compd.* **1996**, *233*, 39.

[17] M. Knecht, H. Ebert, W. Bensch, *J. Alloys Compd.* **1997**, *246*, 166.

[18] C. Gatti, V. R. Saunders, C. Roetti, *J. Chem. Phys.* **1994**, *101*, 10686.

[19] J. A. Platts, S. T. Howard, *J. Chem. Phys.* **1996**, *105*, 4668.

[20] (a) P. L. A. Popelier, Comp. Phys. Comm. **1996**, *93*, 212; (b) P. L. A. Popelier, Comp. Phys. Comm. **1998**, *108*, 180.

[21] Carlo Gatti, An Electron Density Topological Program for Systems Periodic in N (N=0-3) Dimensions. CNR-ISTM; Milano **1999**.

[22] AIM2000: F. W. Biegler-König, J. Schönbohm, D. Bayles, sitio Web: http://gauss.fh-bielefeld.de/aim2000. Visitado el 15 de febrero de 2012.

[23] M. Barzaghi: PAMoC (Version 2002.0), Online User's Manual, CNR-ISTM, Institute of Molecular Science and Technologies, Milano, Italy, 2002. sitio Web: http://www.istm.cnr.it/~barz/pamoc/. Visitado el 15 de febrero de 2012.

[24] AIMAll, 08.05.04: TA Keith (2008), sitio Web: http://aim.tkgristmill.com. Visitado el 15 de febrero de 2012.

[25] Aimpac: http://www.chemistry.mcmaster.ca/aimpac/imagemap/imagemap.htm. Visitado el 15 de febrero de 2012.

[26] Multiwfn: Tian Lu, Feiwu Chen, *J. Comp. Chem.* **2012**, *33*, 580.

[27] M. Kohout, DGrid 4.6, User's Guide (2011). sitio Web:
http://www.cpfs.mpg.de/~kohout/Documents/DGrid-4.6.pdf. Visitado el 15 de febrero de 2012.

[28] Critic: A. Otero-de-la-Roza, M.A. Blanco, A. Martín Pendás, Víctor Luaña, *Comp. Phys. Comm.* **2009**, *180*, 157.

[29] D. Kincaid, W. Cheney, Numerical Analysis, Brooks/Cole Publishing Company 1991.

[30] Y. Aray, J. Rodriguez, D. Vega, Atoms in Molecules: Theory for Exploring the Nature of the Active Sites on Surfaces; C. Matta, R. Boyd, Eds. The Quantum Theory of Atoms in Molecules- From Solid State to DNA and Drug Design; Ed. Wiley-VCH: **2007**; pp. 231-256.

[31] (a) Y. Aray, J. Rodríguez, D. Vega, S. Coll, E. Rodríguez-Arias, F. Rosillo, *J. Phys. Chem.* B **2002**, *106*, 13242; (b) Y. Aray, M. Marquez, J. Rodríguez, S. Coll, Y. Simón-Manso, C. Gonzalez, D. Weitz, *J. Phys. Chem. B* **2003**, *107*, 8946; (c) Y. Aray, M. Marquez, J. Rodríguez, D. Vega, Y. Simón-Manso, S. Coll, C. Gonzalez, D. Weitz, *J. Phys. Chem. B* **2004**, *108*, 2418; (d) Y. Aray, J. Rodríguez, S. Coll, E. Rodríguez-Arias, D. Vega, *J. Phys. Chem. B* **2005**, *109*, 23564; (e) Y. Aray, A. Vidal, J. Rodriguez, M. Grillo, D. Vega, D. Coll *J. Phys. Chem. C* **2009**, *113*, 19545.

[32] C. Katan, P. Rabiller, C. Lecomte, M. Guezo, V. Oisona, M. Souhassoub, *J. Appl. Cryst.* **2003**, *36*, 65.

[33] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C, Cambridge University Press, 1992.

[34] D. Vega, Modelaje de la adsorción de moléculas pequeñas sobre superficies metálicas, Tesis Doctoral, Instituto Venezolano de Investigaciones Científicas, Caracas, Venezuela, **2004**. Apéndice B

[35] DMol^3, user guide, release 3.2, Accelrys Software Inc. **2005**.

[36] GAUSSIAN 09 user's reference, 2011: http://www.gaussian.com/g_tech/g_ur/u_cubegen.htm. Visitado el 15 de febrero de 2012.

[37] M. S. Gordon, M. W. Schmidt, Advances in electronic structure theory: GAMESS a decade later. In C. E. Dykstra, G. Frenking, K. S. Kim, G. E. Scuseria (editors). Theory and Applications of Computational Chemistry: the first forty years. Elsevier, Amsterdam, **2005**; pp. 1167-1189

[38] VASP the GUIDE, Universität Wien, Austria, 2011, sitio Web:
http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html. Visitado el 15 de febrero de 2012.

Artículo 2

D. Vega, D. Almeida "AIM-UC: An application for QTAIM analysis" *Journal of Computational Methods in Sciences and Engineering* **2014**, *14*, 131–136.

DOI 10.3233/JCM-140491

http://iospress.metapress.com/content/9748185r23425nq0/

AIM-UC: Una aplicación para el análisis QTAIM

D. Vega y D. Almeida

Laboratorio de Química Computacional, FACYT, Universidad de Carabobo, Av. Salvador Allende, Edif. FACYT-Química, Bárbula, Edo. Carabobo. Venezuela

Resumen. En este artículo introducimos a AIM-UC, una aplicación gratuita que permite crear gráficos relacionados con la Teoría Cuántica de Átomos en Moléculas (QTAIM). Los datos de entrada son archivos que contienen retículas 3D de la densidad electrónica, en algunos formatos conocidos: CUBE de gaussian, grd de Dmol y CHGCAR de VASP. También tiene soporte para archivos de función de onda de AIMPAC. La aplicación calcula retículas regulares 2D (de densidades electrónicas y sus laplacianas) en un plano posicionado por el usuario. Los puntos críticos del plano se calculan usando interpolación bicúbica y el método de Newton-Raphson, y el campo de gradiente y el grafo molecular, también se generan, utilizando el método de Cash-Karp Runge-Kutta. También permite graficar la densidad electrónica y su laplaciana negativa en el plano seleccionado. Además, para ambas retículas 2D, es posible calcular los mapas de contorno usando el algoritmo "Marching Squares". Esta aplicación también permite encontrar los puntos críticos en el espacio 3D. Las salidas consisten en archivos de texto simple e imágenes en formatos PNG, BMP y EPS (post script encapsulado). AIM-UC fue programado en C++ usando la herramienta de interfaz gráfica de usuario FLTK (fast light toolkit) y OpenGL (open graphics library).

Palabras claves: QTAIM, aplicación gratuita, retícula 2D, retícula 3D.

1. Introducción

La teoría cuántica de átomos en moléculas (QTAIM) permite representar enlaces químicos,

predecir su multiplicidad y delimitar el espacio ocupado por cada átomo en moléculas y sólidos cristalinos, a través del análisis topológico de la densidad de carga electrónica. Es también posible representar concentraciones de electrones (análogo al par sin compartir de Pauling) mediante el análisis de la topología de la laplaciana de la densidad electrónica. QTAIM [1-2] fue desarrollada por Richard Bader [3] en la década del 70 y ha sido útil en el análisis de propiedades de sistemas con densidades electrónicas ab-initio [4-6] y experimentales [7-9].

Existen varias aplicaciones para análisis QTAIM, entre ellas gratuitas de código abierto: Aimpac [10], Dgrid [11], Critic [12], PAMoC [13] sin interfaz gráfica y Multiwfn [14] con una interfaz gráfica limitada. Otras con buenas interfaces gráficas, pero no gratuitas: AIMAll [15] y AIM2000 [16]. Aclarando que la lista de aplicaciones citadas no es exhaustiva. En nuestro laboratorio desarrollamos una aplicación gratuita con una interfaz gráfica amigable, AIM-UC, la cual genera algunos gráficos (campo de gradiente, grafo molecular y mapa de contornos) relacionados con QTAIM. Además permite graficar la densidad electrónica y su laplaciana negativa. Aunque en el futuro cercano, no hay planes para soltar el código fuente, están disponibles versiones para algunos sistemas operativos: Windows de 32 y 64 bits de Microsoft, y los siguientes Linux de 64-bits: Centos 6.3, Ubuntu 12.10 y Xubuntu 12.10.

AIM-UC puede leer datos de archivos que contienen retículas de densidad electrónica: CUBE de Gaussian [17], GRD de DMol [18], CHGCAR de VASP [19] entre otros (ver el manual de usuario), y, añadido recientemente, archivos de función de onda de AIMPAC [10]. Todas las imágenes mostradas por AIM-UC pueden guardarse en formatos PNG o BMP, y el mapa de contorno 2D y las líneas de gradiente también pueden guardarse en formato EPS. La aplicación también permite guardar archivos de texto simple con información acerca de los puntos críticos.

Una característica notable de esta aplicación es el conjunto de herramientas para posicionar el plano (retícula 2D contenida en un paralelogramo) que incluyen desplazamiento y rotación. También las coordenadas del origen y de los dos vectores (no necesariamente ortogonales) que forman el paralelogramo pueden introducirse directamente, o el plano puede

ser posicinado de modo que contenga tres puntos (escogidos entre los núcleos atómicos y los puntos críticos). La fig. 1 muestra la ventana que contiene estas herramientas.

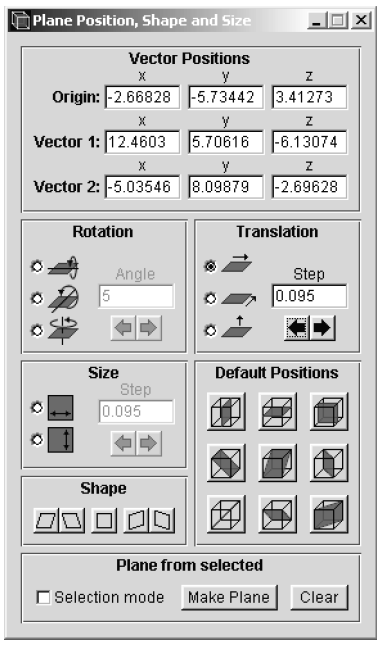

Fig. 1. Ventana de la aplicación que contiene las herramientas para posicionar el plano (2D grid).

Otra característica importante es la capacidad de trabajar con retículas 3D inmensas. La retícula se almacena la memoria con asignación dinámica, usando un apuntador a un vector de apuntadores a vectores de apuntadores a datos (un apuntador triple), asignando un máximo de memoria continua igual al tamaño del dato multiplicado por el número máximo de puntos en una dimensión. En un computador personal con 4 GB RAM, la aplicación puede trabajar con una retícula de 900 × 900 × 900 puntos.

Además, soporta sistemas periódicos:Los puntos críticos no se replican cuando están situados en las caras de la celda periódica, y el plano se grafica correctamente aun si parte de éste está fuera de la celda unidad.

AIM-UC puede obtenerse de sitio web http://alfa.facyt.uc.edu.ve/quimicomp/

2. Construcción de las retículas 2D

Las retículas 2D de la densidad electrónica y su Laplaciana se construyen usando interpolación tricúbica de Lagrange [20], El valor interpolado (P) se da en la ec. 1.

$$P=\sum_{k_1=0}^{3}\sum_{k_2=0}^{3}\sum_{k_3=0}^{3} w_{k_1}\cdot w_{k_2}\cdot w_{k_3}\cdot f_{k_1k_2k_3} \qquad (1)$$

donde w_{k_i} son los polinomios base de Lagrange (que dependen de las coordenadas del punto a ser evaluado) y $f_{k_1k_2k_3}$ son los valores de los puntos de la retícula que rodean al punto a ser evaluado.

Cuando los datos de entrada provienen de un archivo de función de onda de AIMPAC [10], ambas retículas se calculan analíticamente. La retícula está dentro de un paralelogramo definido por el origen de coordenadas (**o**) y dos vectores linealmente independientes ($\mathbf{v}_1$ and $\mathbf{v}_2$), no necesariamente ortogonales. La densidad y su laplaciana se evalúan en las coordenadas del punto (**p**) dadas por la ec. 2.

$$\mathbf{p}=\mathbf{o}+(i/N_1)\,\mathbf{v}_1+(j/N_2)\,\mathbf{v}_2 \qquad (2)$$

donde N_1 and N_2 son los números de intervalos en los cuales los vectores $\mathbf{v}_1$ y $\mathbf{v}_2$ se dividen, y los índices i y j van desde 0 a N_1 y 0 a N_2, respectivamente.

Si el valor de la densidad electrónica es mayor que "Nuclear density", se utiliza una interpolación tricúbica de logaritmos de la densidad [20]. Así se previene que el polinomio de interpolación oscile cerca del núcleo, donde la densidad electrónica incrementa apreciableente. Un valor adecuado para "Nuclear density" es la mitad de la máxima densidad de un átomo de hidrógeno. Este valor no tiene efecto en los archivos de función de onda de AIMPAC.

3. Puntos críticos

La aplicación permite encontrar los puntos críticos em el espacio y en retículas 2D, usando el método de Newton-Raphson [21], excepto cerca de las posiciones nucleares (máximos), donde este método falla debido al valor elevado del módulo del vector gradiente. Para encontrar los máximos, el algoritmo busca los puntos de la retícula que tengan densidad mayor o igual que todos sus vecinos cercanos (4 vecinos en 2D y 6 en 3D), entonces se selecciona el punto con mayor valor de densidad en la dirección de los vecinos, pero a la mitad de la distancia. Los máximos son los puntos obtenidos cuando el proceso se repite varias veces, siempre reduciendo la distancia, moviéndose en las mismas direcciones iniciales y usando la interpolación de logaritmos de densidades. En el caso de un archivo de función de onda de AIMPAC, los puntos críticos máximos 3D (cercanos al núcleo), son justo las coordenadas de los núcleos.

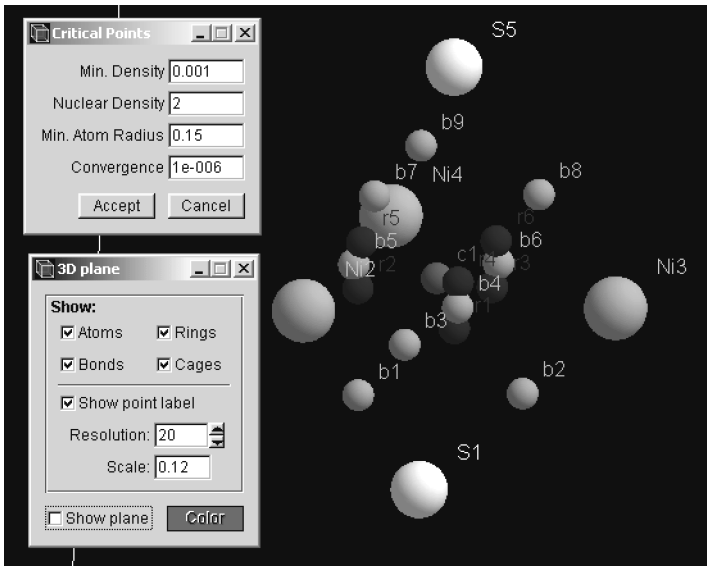

Fig. 2. Puntos críticos encontrados en un agregado Ni_3S_2. Los núcleos se azufre son las esferas grandes de arriba y abajo, los de níquel son las tres esferas grandes del medio. Hay un punto crítico de caja (etiquetado como c1), seis puntos de anillo (esferas oscuras pequeñas) arriba y abajo de cada punto crítico de enlace Ni-Ni, y nueve puntos de enlace (etiquetados desde b1 a b9).

Para los puntos críticos remanentes, debido a alta velocidad de cálculo del método de interpolación, se utiliza una búsqueda sistemática. Algunos puntos de la retícula son los puntos iniciales para el método de Newton-Raphson y se permite un desplazamiento limitado. En el caso de un archivo de función de onda de AIMPAC, el método usa como punto inicial el promedio de coordenadas de cada par, trío y cuarteto de átomos. La fig. 2 muestra los puntos críticos para el agregado Ni_3S_2, encontrados en una retícula de densidad con 163 × 169 × 188 puntos, calculados al nivel de teoría DFT-PBE/DNP (optimización de geometría completa) utilizando el programa DMol^3 [18].

4. Campos de gradiente e isolíneas

Los caminos de gradiente de las retículas 2D se construyen usando el método de Cash-Karp Runge-Kutta [21]. Las líneas que representan las cuencas atómicas comienzan en puntos localizados en un círculo alrededor de cada máximo, siguiendo el gradiente negativo. El grafo molecular se obtiene de los dos caminos que parten de ambos lados de cada punto crítico de ensilladura en la dirección de los vectores propios de la matriz hessiana (evaluada en el punto de ensilladura) con valor propio positivo, siguiendo el gradiente. Las líneas que delimitan las cuencas atómicas también se obtienen de los puntos de ensilladura, pero en la dirección del otro vector propio, y siguiendo el gradiente negativo.

No todos los máximos del plano corresponden a átomos, ni todos los puntos de ensilladura corresponden a puntos críticos de enlace. La aplicación calcula la distancia entre los puntos críticos del plano y el espacio, ocultando los caminos de enlace y de gradiente de la cuenca cuando no hay coincidencia de puntos críticos. La aplicación también permite al usuario ocultar o mostrar las líneas, cambiar su color, tipo y ancho.

Las isolíneas de la densidad de carga electrónica y su laplaciana se construyen con el algoritmo "Marching Squares" [22]. Por cada cuadro de la retícula se comparan los valores en los vértices con el isovalor. En caso de encontrar aristas con un vértice con valor mayor y el otro con valor menor que el isovalor, el punto de intercepción se calcula por interpolación

lineal, y estos se unen para formar la isolínea correspondiente. El mapa de contornos de la laplaciana de la densidad electrónica se muestra en la fig. 3, para una retícula 2D con 600 × 400 puntos que contiene todas las posiciones atómicas de la molécula *p*-nitroanilina. El rectángulo gris, señalado por una flecha, se amplifica en la parte inferior derecha de la imagen. Las líneas consisten en segmentos rectos. El cálculo de la retícula inicial fue llevado a cabo usando el programa Gamess-US [23] al nivel HF/6-31G(2p,2d) y el espaciamiento de la retícula fue 0.1 bohr.

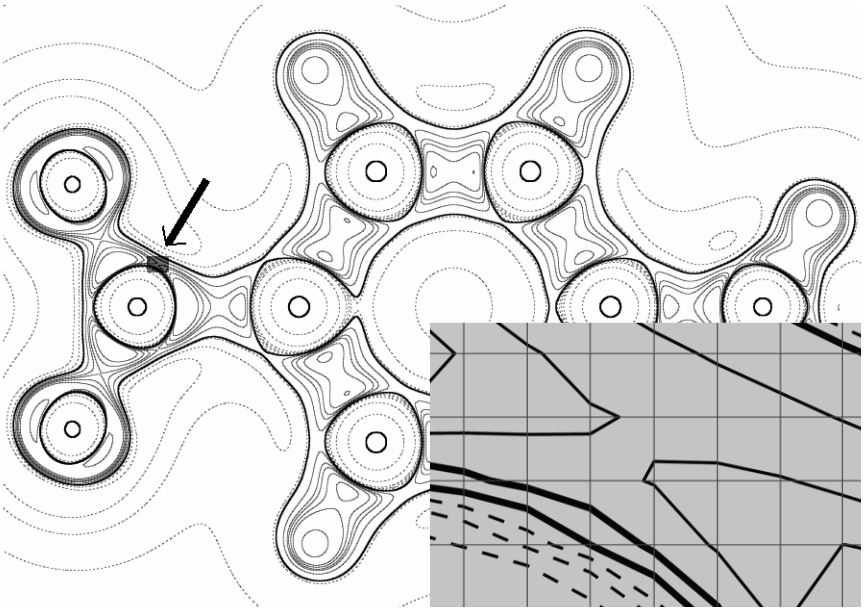

Fig. 3. Mapa de contornos de la molécula *p*-nitroanilina. Las líneas gruesas, las punteadas y las delgadas corresponden a isovalores de valor cero, positivo y negativo respectivamente. El rectángulo gris claro, abajo a la derecha, es una ampliación para diferenciar los puntos de la retícula (unidos por líneas grises).

5. Comentarios finales

El propósito inicia de AIM-UC fue realizar dibujos 2D y además calcular los puntos críticos en 3D para determinar si los enlaces y caminos de gradiente se muestran. Sin embargo, actualmente estamos trabajando en dibujos 3D, adicionando la topología de la densidad electrónica (caminos de enlace y superficies interatómicas) y su laplaciana (grafo atómico), e

incluyendo las cargas atómicas calculadas por integración numérica.

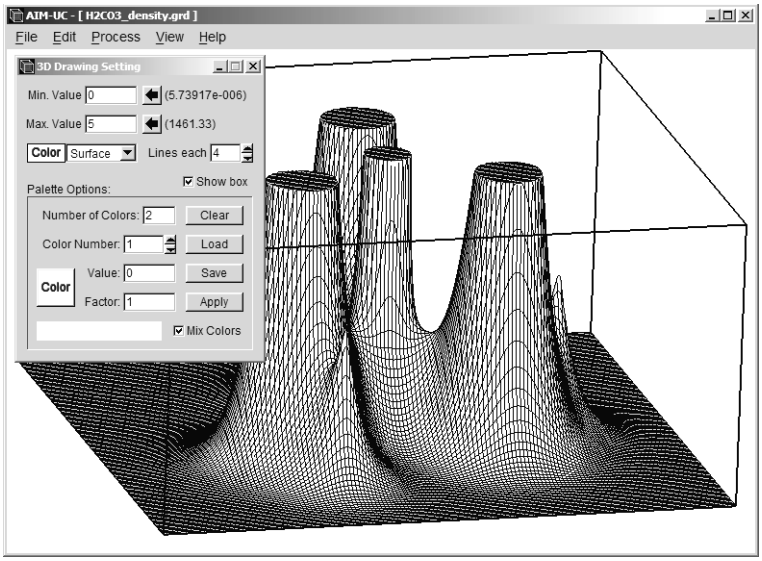

Fig. 4. Gráfico de densidad electrónica en un plano que contiene todas las posiciones atómicas de la molécula de ácido carbónico. Puede cambiarse los colores de la superficie, líneas y del fondo, y el número de líneas.

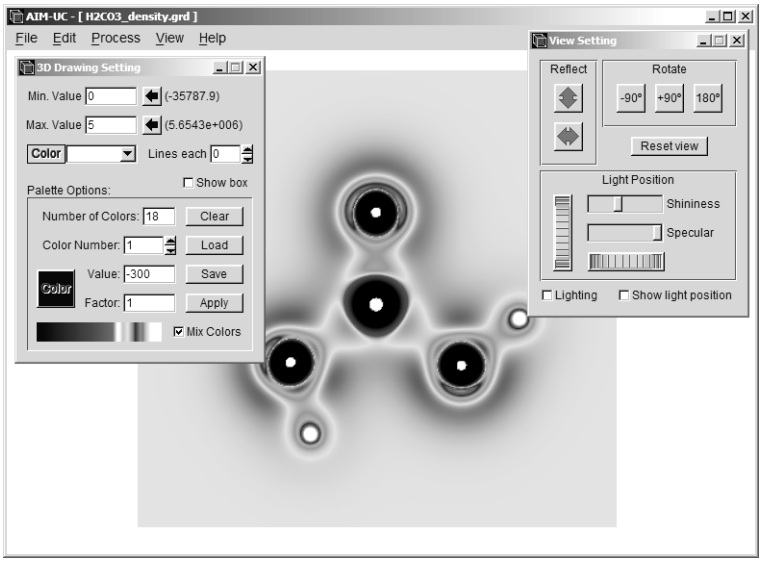

Fig. 5. Laplaciana de la densidad electrónica para la molécula de ácido carbónico, sin efecto de iluminación y coloreada por valor.

Excepto por el rectángulo añadido a la fig. 3, todas las figuras son capturas de la ventana de la aplicación. Otras capacidades de AIM-UC se muestran en las fig. 4 y 5. El gráfico de la densidad electrónica de la molécula de ácido carbónico en un plano que contiene todos los átomos (fig. 4), y la laplaciana negativa (coloreada de acuerdo a su valor) de la densidad electrónica de la misma molécula en el mismo plano (fig. 5), la retícula de entrada fue calculada al nivel DFT-PBE/DNP usando el programa DMol[3].

Las versiones de AIM-UC fueron compiladas bajo el sistema operativo Microsoft Windows 64-bit usando TDM-GCC [24], Windows 32-bit con MinGW [25] y Linux con GCC. También usamos las bibliotecas externas FLTK 1.x [26] para construir la interfaz de usuario, OpenGL para gráficos y libpng [27] para generar archivos de imágenes PNG. Para más información lea el manual de usuario de AIM-UC, el cual puede obtenerse del sitio web mencionado al final de la sección introducción.

Agradecimientos

Este trabajo fue financiado por el proyecto G2005000424 del Fondo Nacional de Ciencias y Tecnología (Fonacit) de Venezuela.

References

[1] R. F. W. Bader, *Atoms in molecules-a Quantum Theory*, Clarendon Press, Oxford, U.K., 1990;

[2] R. F. W. Bader, *Chem Rev* **91** (1991), 893-928.

[3] Obituary in Canadian Chemical News: http://www.accn.ca/index.php?ci_id=3370&la_id=1. Visitado el 15 de abril de 2013.

[4] F. Cortés-Guzmán, y R. F. W. Bader. *Coord. Chem. Rev.* **249** (2005), 633-662.

[5] D. Coll, F. Delbecq, Y. Aray, y P. Sautet.. *Phys. Chem. Chem. Phys.* **13** (2011): 1448-1456.

[6] M Jabłoński y M Palusiak. *J. Phys. Chem. A.* **116** (2012), 2322-2332.

[7] L. J. Farrugia, C. Evans, H. M. Senn, M. M. Hänninen, y R. Sillanpää. *Organometallics*, **31** (2012), 2559-2570.

[8] P. Sledz, R. Kaminski, M. Chruszcz, M. D. Zimmerman, W. Minor y K. Wozniak. *Acta Cryst.* **B66** (2010), 482-492.

[9] Vladimir V. Zhurov, Elizabeth A. Zhurova, y A. Alan Pinkerton. *Inorg. Chem.*, **50** (2011), 6330-6333.

[10] Aimpac: http://www.chemistry.mcmaster.ca/aimpac/imagemap/imagemap.htm
Visitado el 15 de abril de 2013.

[11] M. Kohout, DGrid 4.6, User's Guide, Max Planck Institute for Chemical Physics of Solids, Dresden, Germany, 2011. Sitio web: http://www.cpfs.mpg.de/~kohout/Documents/DGrid-4.6.pdf. Visitado el 15 de abril de 2013.

[12] Critic: A. Otero-de-la-Roza, M. A. Blanco, A. Martín Pendás, V. Luaña, *Comp. Phys. Comm*. **180** (2009), 157-166.

[13] M. Barzaghi: PAMoC (Version 2002.0), Online User's Manual; CNR-ISTM, Institute of Molecular Science y Technologies: Milano, Italy, 2002. Sitio web:
http://www.istm.cnr.it/~barz/pamoc/. Visitado el 15 de abril de 2013.

[14] Multiwfn: T. Lu y F. Chen, *J. Comp. Chem.*, **33** (2012), 580-592.

[15] AIMAll (Version 12.06.03), Todd A. Keith, TK Gristmill Software, Overland Park KS, USA, 2012, Sitio web: http://aim.tkgristmill.com. Visitado el 15 de abril de 2013.

[16] AIM2000: F. W. Biegler-König, J. Schönbohm, D. Bayles, Sitio web:
http://gauss.fh-bielefeld.de/aim2000. Visitado el 15 de abril de 2013.

[17] GAUSSIAN 09 user's reference, 2011: http://www.gaussian.com/g_tech/g_ur/u_cubegen.htm. Visitado el 15 de abril de 2013.

[18] DMol3, user guide, release 3.2, Accelrys Software Inc. 2005.

[19] VASP the GUIDE, Universität Wien, Austria, 2011,
http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html. Visitado el 15 de abril de 2013.

[20] D. Vega, Y. Aray, J. Rodríguez, *J. Comp. Chem.*, **33** (2012), 2526-2531.

[21] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1992.

[22] Una descripción de este algoritmo puede encontrarse en el sitio web:
http://users.polytech.unice.fr/~lingrand/MarchingCubes/algo.html. Visitado el 15 de abril de 2013.

[23] M. S. Gordon, M. W. Schmidt, Advances in electronic structure theory: GAMESS a decade later, in: *Theory and Applications of Computational Chemistry: The First Forty Years*. C. E. Dykstra, G. Frenking, K. S. Kim, G. E. Scuseria, eds, Elsevier, Amsterdam, 2005. pp. 1167-1189

[24] TDM-GCC es una suite de compiladores para Windows basada en GCC y MinGW:
http://tdm-gcc.tdragon.net/. Visitado el 15 de abril de 2013.

[25] MinGW es un conjunto de herramientas de programación de código abierto para Windows: http://www.mingw.org/

[26] FLTK es una plataforma cruzada, C++ herramientas de interfaz gráfica de usuario, para UNIX/Linux (X11), Microsoft Windows, y MacOS X: http://www.fltk.org/. Visitado el 15 de abril de 2013.

[27] libpng es la biblioteca PNG de referencia oficial: http://www.libpng.org/pub/png/libpng.html. Visitado el 15 de abril de 2013.

APÉNDICES

(artículos originales)

C Library for Topological Study of the Electronic Charge Density

David Vega,*[a] Yosslen Aray,[b] and Jesús Rodríguez[b]

The topological study of the electronic charge density is useful to obtain information about the kinds of bonds (ionic or covalent) and the atom charges on a molecule or crystal. For this study, it is necessary to calculate, at every space point, the electronic density and its electronic density derivatives values up to second order. In this work, a grid-based method for these calculations is described. The library, implemented for three dimensions, is based on a multidimensional Lagrange interpolation in a regular grid; by differentiating the resulting polynomial, the gradient vector, the Hessian matrix and the Laplacian formulas were obtained for every space point. More complex functions such as the Newton–Raphson method (to find the critical points, where the gradient is null) and the Cash–Karp Runge–Kutta method (used to make the gradient paths) were programmed. As in some crystals, the unit cell has angles different from 90°, the described library includes linear transformations to correct the gradient and Hessian when the grid is distorted (inclined). Functions were also developed to handle grid containing files (grd from DMol® program, CUBE from Gaussian® program and CHGCAR from VASP® program). Each one of these files contains the data for a molecular or crystal electronic property (such as charge density, spin density, electrostatic potential, and others) in a three-dimensional (3D) grid. The library can be adapted to make the topological study in any regular 3D grid by modifying the code of these functions.
© 2012 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23083

Introduction

The "*quantum theory of atoms in molecules*" (QTAIM) developed by Bader et al.[1–8] is very useful to obtain the chemical information from the charge density. QTAIM is a firm, rigorous, and quantum mechanically well-defined theory based on observables such as the electron density or energy density fields. Most modern theories of bonding are based, in one way or another, on the partition of charge (or electronic density) among the different nuclear centers under study, usually according to Mulliken, that is, projected density of states in solid-analysis. In this way, an important amount of the interpretative models of chemical behavior are based on concepts that are known to be poorly defined and giving answers extremely dependent on a whole hierarchy of approximations.[6] QTAIM provides a quantitative link between the total electron density (regardless of how it was generated: calculated or experimental) and important physical properties of a molecule, bypassing the wave function in the analysis. In contrast, QTAIM is a methodology independent from the orbital concept. In particular, it provides a rigorous definition of chemical bond and geometrical structure for all types of molecules and solids and it has proven to be useful in the analysis of physical properties of insulators, pure metals, and alloys.[4–6] High-quality experimental densities of minerals,[9,10] covalent,[11] metallic,[12] and molecular crystals[13,14] have been analyzed in terms of QTAIM concepts. Furthermore, QTAIM-calculations on simple metals,[15] alloys, and intermetallic phases[16,17] have also been reported as well as on molecular,[18,19] covalent, and ionic crystals.[6,11] Software packages that calculate and draw gradient path, laplacian, and density isolines has been published,[20–28] using similar computational methods; however, only few of them are open source code (Aimpac,[25] Multiwfn,[26] DGrid,[27] and Critic[28]).

The topological study consists in determining and characterizing critical points, bond paths, zero flux surfaces, gradient maps, and atomic basins. The critical points are ones where the gradient is null (among these points exist local minima and maxima, and saddle points) and are fundamental to the topological study. The local maxima usually correspond to the atomic nuclei positions, the local minima are cage critical points; and there are two kinds of saddle points, minimum in one direction and maximum in the perpendicular plane (bond critical points), and maximum in one direction and minimum in the perpendicular plane (ring critical points). A bond and interatomic surface representations are obtained by drawing gradient paths close to a bond critical point. To find these points, it is necessary to obtain the gradient vector, $\mathbf{g}(\mathbf{r})$ and the Hessian matrix, $\mathbf{H}(\mathbf{r})$ in any space point $\mathbf{r}$ (using the Newton–Raphson method, later described).

[a] *D. Vega*
Dpto. de Química, Facultad de Ciencias y Tecnología, Universidad de Carabobo, Ciudad Universitaria, Bárbula, Valencia, Venezuela
E-mail: dvega@uc.edu.ve
[b] *Y. Aray, J. Rodríguez*
Centro de Química, IVIC, Apartado 21827, Caracas 1020 A, Venezuela
Contract/grant sponsor: Fondo Nacional de Ciencias y Tecnología (Fonacit) of Venezuela; Contract/grant number: G-2005000424.
© 2012 Wiley Periodicals, Inc.

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \mathbf{g}(\mathbf{r}) = \nabla f(\mathbf{r}) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} \quad \mathbf{H}(\mathbf{r}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial x \partial z} & \frac{\partial^2 f}{\partial y \partial z} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix} \tag{1}$$

To replace gradient and Hessian calculation routines in software packages for finding critical points and their properties, the library began to build at early 2000s using Fortran 77. For developing our own programs comprehensively, later on, the library was translated to C language. This C library content was early reported,[29] not in detail, but including some calculation examples. Also, it has been used as a tool in studies of ours group.[30] In this work, we describe exhaustively the C library, providing the source code, method of use and some additional functions for reading grids in different formats.

The library contains functions to calculate the gradient vector and the Hessian matrix, and other functions useful for the topological study.

Approximation to the Function and Derivatives

Starting with the Lagrange polynomial approximation formula,[31,32] eq. (2):

$$P(x) = \sum_{k=0}^{n-1} \frac{\prod_{j=0,\, j\neq k}^{n-1} (x - x_j)}{\prod_{j=0,\, j\neq k}^{n-1} (x_k - x_j)} f_k \tag{2}$$

where n is the number of points (x_j, f_j) used in the interpolation and $P(x)$ is a polynomial of degree $n-1$ that passes through the

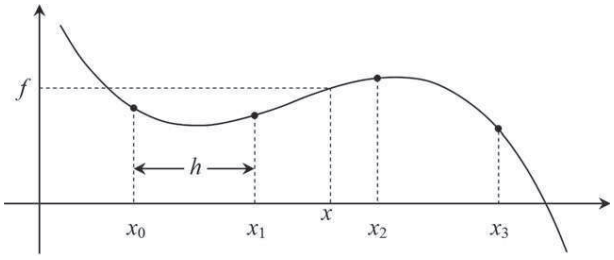

Figure 1. The Lagrange polynomial passes through the points (black dots). When the polynomial is evaluated in another point (x), an approximate value (f) of the function is obtained. When possible, in a piecewise interpolation, it is convenient to choose the points array such that the x value is inside the central interval.[29]

n points. If the x_j values are equally spaced[29] (Fig. 1), then $x_j = x_0 + j\,h$, where h is the distance among the points.

Defining $s = (x - x_\alpha)/h$, with a constant α index (it is convenient to choose α such that x_α and $x_{\alpha+1}$ are the central points of the array: n is even and $\alpha = n/2-1$), solving for x ($x = x_\alpha + s{\cdot}h$) and substituting in the Lagrange polynomial, eq. (2):

$$P = \sum_{k=0}^{n-1} w_{k,n}(s) \cdot f_k \tag{3}$$

where

$$w_{k,n}(s) = \frac{\prod_{j=0,\, j\neq k}^{n-1} (\alpha - j + s)}{\prod_{j=0,\, j\neq k}^{n-1} (k - j)} \tag{4}$$

$w_{k,n}(s)$ is a $n-1$ degree polynomial in s.

To obtain the approximations of the function derivatives in x, the $w_{k,n}(s)$ is differentiated with respect to s, as shown:

$$\frac{dP}{dx} = \frac{dP}{ds}\frac{ds}{dx} = \frac{1}{h}\frac{dP}{ds} = \frac{1}{h}\sum_{k=0}^{n-1} \frac{dw_{k,n}(s)}{ds} f_k \tag{5}$$

For the higher order derivatives,

$$\frac{d^v P}{dx^v} = \frac{1}{h^v}\sum_{k=0}^{n-1} w_{k,n}^{(v)}(s) \cdot f_k \tag{6}$$

where

$$w_{k,n}^{(v)}(s) = \frac{d^v w_{k,n}(s)}{ds^v} \tag{7}$$

In the interpolation and approximation of the derivatives (in the developed library), the used polynomials $w_{k,n}^{(v)}(s)$ were calculated for four points ($n = 4$). See Table 1.

Expressing the Lagrange polynomial as a sum of products of f_k and the weight $w_{k,n}(s)$, eq. (2), permits to obtain the derivatives in a straightforward way, eq. (5). Also, calculating the tensor product[31] of the weights $w_{k,n}(s)$, an approximation for a multidimensional regular array can be figured out. For a three-dimensional (3D) grid, for each dimension, the value of s_i is calculated ($s_1 = (x - x_\alpha)/h_1$, $s_2 = (y - y_\alpha)/h_2$), $s_3 = (z - z_\alpha)/h_3$), then the $w_{k_i,n_i}(s_i)$ polynomials are evaluated. These polynomials multiply the function values for each grid point ($f_{k_1k_2k_3}$):

$$P = \sum_{k_1=0}^{n_1-1}\sum_{k_2=0}^{n_2-1}\sum_{k_3=0}^{n_3-1} w_{k_1,n_1}(s_1) \cdot w_{k_2,n_2}(s_2) \cdot w_{k_3,n_3}(s_3) \cdot f_{k_1k_2k_3} \tag{7}$$

where n_1, n_2, and n_3 are the numbers of points used in the interpolation along each dimension.

Differentiating the resulting polynomial with respect to s_i, the approximation to any of the derivatives, with respect to x, y, z variables, is obtained.

$$\frac{\partial P^{v_1+v_2+v_3}}{\partial x^{v_1}\partial y^{v_2}\partial z^{v_3}} = \frac{1}{h_1^{v_1}h_2^{v_2}h_3^{v_3}}\sum_{k_1=0}^{n_1-1}\sum_{k_2=0}^{n_2-1}\sum_{k_3=0}^{n_3-1} w_{k_1,n_1}^{(v_1)}(s_1) \cdot w_{k_2,n_2}^{(v_2)}(s_2) \cdot w_{k_3,n_3}^{(v_3)}(s_3) \cdot f_{k_1k_2k_3} \tag{8}$$

Equation (8) is equivalent to eq. (7), when $v_1 = v_2 = v_3 = 0$, understanding that $w_{k,n}^{(0)}(s) = w_{k,n}(s)$.

The developed library principally consist in routines that, given the function values in a 3D regular grid, calculate the approximate value of the function, gradient vector, and Hessian matrix, in any space point inside the grid limits. These routines are implementations of eq. (8) with $n_1 = n_2 = n_3 = 4$ (tricubic Lagrange interpolation). Different from trilinear

Table 1. $w_{k,n}^{(v)}(s)$ polynomials defined in eqs. (7) and (4), with $n = 4$ y $\alpha = 1$, and $w_{k,n}^{(0)}(s) = w_{k,n}(s)$.

	$k = 0$	$k = 1$	$k = 2$	$k = 3$
$v = 0$	$(-s^3 + 3s^2 - 2s)/6$	$(s^3 - 2s^2 - s + 2)/2$	$(-s^3 + s^2 + 2s)/2$	$(s^3 - s)/6$
$v = 1$	$(-3s^2 + 6s - 2)/6$	$(3s^2 - 4s - 1)/2$	$(-3s^2 + 2s + 2)/2$	$(3s^2 - 1)/6$
$v = 2$	$-s + 1$	$3s - 2$	$-3s + 1$	s

interpolation,[33] this allows for approximations of second-order derivatives (see Table 1).

Inclined grid

The "regular" term for the grid, means that the space between the points (h_i) is constant along any particular dimension; however, it can be different along the other dimension. Moreover, the grid can be inclined (Fig. 2). This means that, at least

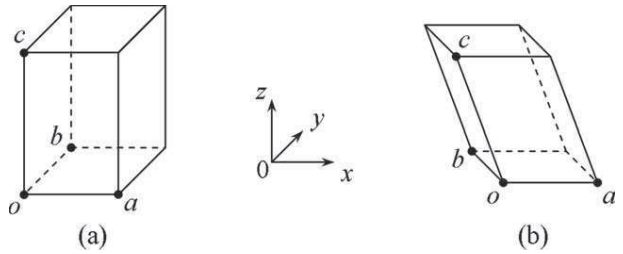

Figure 2. a) No inclined grid, *o* is situated in the origin, *a* in the *x* axis, *b* in the *y* axis, and *c* in the *z* axis. b) Inclined grid, *o* is situated in the origin, *a* in the *x* axis, *b* in the *xy* plane and *c* could be in any place.

one of the angles among the axes of the dimensions is different from 90° (this is usual for the unit cell of many crystalline solids). In this case, we always take one axis parallel to the *x* axis and another parallel to the *xy* plane.

To calculate the evaluated gradient in a space point **r** in an inclined grid, it is necessary to perform two linear transformations,[34] first, the point coordinates in the inclined coordinated system (**r**′) must be calculated, $\mathbf{Ar} = \mathbf{r}'$, where **A** is the matrix that transforms the ordinary (no inclined) coordinates to inclined coordinates. Then, calculating the derivatives, using eq. (8), the gradient vector (**g**′) is determined. Finally, the gradient is transformed back to the ordinary coordinates, multiplying by the transpose of **A**, $\mathbf{g} = \mathbf{A}^T\mathbf{g}'$.

To calculate the Hessian matrix (**H**), it is also necessary to transform the **r** point to the inclined coordinates system. Once **H**′ is calculated (using eq. (8) to get the second-order derivatives), **H**′ is finally transformed back to the ordinary coordinates: $\mathbf{H} = \mathbf{A}^T\mathbf{H}'\mathbf{A}$.

These equations are quite general, when a linear transformation exists from a coordinated system to another, however, in the developed library; the only case considered is when the inclined coordinated system satisfies the given conditions (Fig. 2). In this case, the transform matrix is always an upper triangular matrix.

Approximation to the function logarithm

The electronic charge density has particular characteristics, for example, its value is never negative, and it has the exponential behavior close to the atomic nuclei. The great increment of the density value causes that the interpolated polynomial oscillates producing negative values in the neighborhood of the nuclei (see Fig. 3b). By changing the grid value ($f_{k_1k_2k_3}$) for its logarithm in the previously described interpolation, eq. (7), and finally taking the antilogarithm of the interpolated value (P), it is possible to avoid the oscillation condition.

In the derivative cases, eqs. (9) and (10) are used.

$$\frac{\partial f}{\partial u} = \frac{\partial p}{\partial u} e^p \tag{9}$$

$$\frac{\partial^2 f}{\partial u \partial v} = \left(\frac{\partial^2 p}{\partial u \partial v} + \frac{\partial p}{\partial u}\frac{\partial p}{\partial v}\right) e^p \tag{10}$$

where u and v are x, y, or z, and p is the interpolation of the logarithm. The $\partial p/\partial u$, $\partial p/\partial v$, and $\partial^2 p/\partial u \partial v$ values are the first-order and second-order derivative approximations obtained by the method when the logarithms of the grid values are used.

Methods Requiring the Gradient Vector and the Hessian Matrix

The following methods are used in the topological study and require the gradient vector and the Hessian matrix.

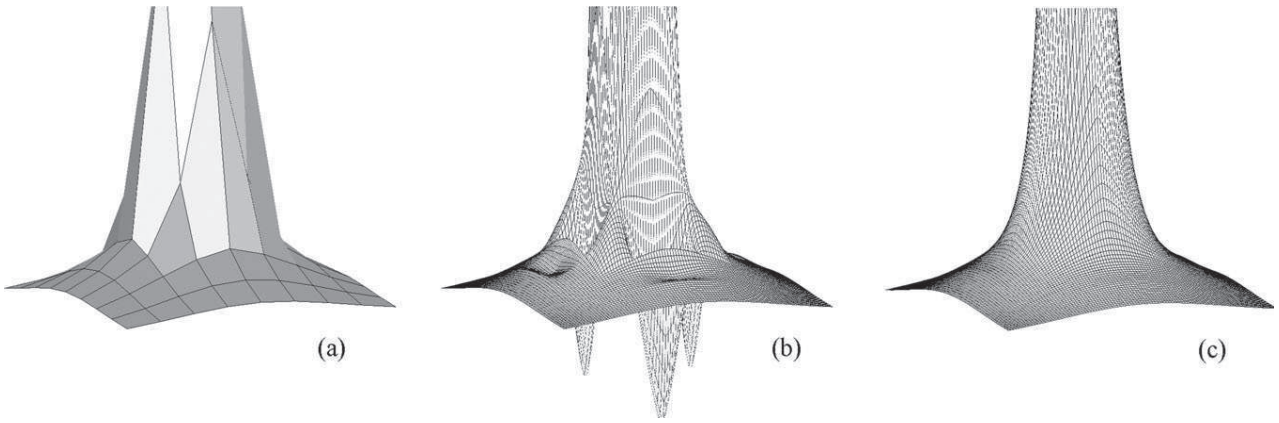

Figure 3. Electronic charge density plots in a plane that contains the sulfur atom of the dibenzotiophene molecule. The plane contains 9 × 9 grid points; the distance between points is 5×10^{-12} m. a) Without interpolation. b) Using polynomial interpolation. c) Using logarithm interpolation.

Newton–Raphson method

Due to the fact that the evaluated function gradient at a critical point is equal to the null vector (**0**), the way to calculate the point coordinates is to solve the equation $\bigtriangledown\rho(\mathbf{r}_{crit}) = \mathbf{0}$. An alternative way to solve this equation is to use the Newton–Raphson method.[33] The multidimensional scalar function $\bigtriangledown\rho(\mathbf{r})$ evaluated at the point $\mathbf{r} = \mathbf{r}_i + \mathbf{h}$ is expanded in Taylor series:

$$\nabla\rho(\mathbf{r}) = \nabla\rho(\mathbf{r}_i) + \mathbf{H}_i\mathbf{h} + \text{higher order terms} \quad (9)$$

where $\mathbf{H}_i$ is the Hessian matrix $\mathbf{H}(\mathbf{r})$ (the Jacobian of $\bigtriangledown\rho$ $(\mathbf{r})$) valuated at $\mathbf{r}_i$.

By neglecting the higher order terms in the Taylor series, eq. (9), setting $\bigtriangledown\rho$ $(\mathbf{r}) = \mathbf{0}$ and solving for **h**:

$$\mathbf{h} = -\mathbf{H}_i^{-1}\nabla\rho(\mathbf{r}_i) \quad (10)$$

we get the shift vector (**h**).

If the function $\rho(\mathbf{r})$ is quadratic then the **h** vector starts in the $\mathbf{r}_i$ point and ends in the critical point. In general, the function is not quadratic so a new $\mathbf{r}_{i+1}$ point is always calculated using:

$$\mathbf{r}_{i+1} = \mathbf{r}_i + t\,\mathbf{h} \quad (11)$$

where t is a small value, lower than 1.

The calculation according to eqs. (10) and (11) is iterated until $\bigtriangledown\rho(\mathbf{r})$ is equal to the vector **0** or has a small norm. Then, $\mathbf{r}_i$ is a critical point or one very close to it.

In the developed algorithm, the norm of **h** has an upper bound. When $|\mathbf{h}|$ is greater than the bound, t is calculated so that $|t\,\mathbf{h}|$ is equal to the bound (in other case $t = 1$). For each iteration, the bound is decreased by means of a geometric progression, in which the ratio must be lower than 1. Doing this, the large oscillation near to the critical points is avoided, when the gradient has a high norm value in the critical point neighborhood. The maximum number of iterations and the path length are defined by the user.

Fifth-order Cash–Karp Runge–Kutta Method

The gradient path necessary for the molecular graph and interatomic surface construction are solutions of the differential eq. (12).[1]

$$d\mathbf{r}(s)/ds = \nabla\rho(\mathbf{r}(s)) \quad (12)$$

Equation (12) solution is a parametric curve in R^3 which is unique when an initial **r** value is given. A numerical solution is obtained with the fifth-order Cash–Karp Runge–Kutta method [33]. The general form of this method is

$$\mathbf{r}_{n+1} = \mathbf{r}_n + c_1\mathbf{k}_1 + c_2\mathbf{k}_2 + c_3\mathbf{k}_3 + c_4\mathbf{k}_4 + c_5\mathbf{k}_5 + c_6\mathbf{k}_6 \quad (13)$$

where $\mathbf{r}_n = (x_n, y_n, z_n)$ and the $\mathbf{k}_j$ values with a little stepsize h are:

$$\mathbf{k}_1 = h\nabla\rho(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_n}$$
$$\mathbf{k}_2 = h\nabla\rho(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_n+b_{21}\mathbf{k}_1}$$
$$\cdots$$
$$\mathbf{k}_6 = h\nabla\rho(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_n+b_{61}\mathbf{k}_1+\cdots+b_{65}\mathbf{k}_5}$$

The particular values of the various constants (c_j, b_{ij}) can be found in the "*Numerical Recipes*."[33] Choosing the stepsize h as an adequate little value, the set of $\mathbf{r}_n$ points is adjusted closely enough to the gradient path.

Derivative Discontinuities

The derivative discontinuities of piecewise Lagrange polynomials do not allow finding some critical points located on the faces of cube grid. Conversely, the spline interpolation assures the derivative continuities, but it requires an array to store the second derivatives and solving several equation systems involving all grid data.[33] To save calculation time and memory storage, we prefer to use the Lagrange interpolation, instead of spline method and solving the discontinuity problem as explained later.

Four grid points are always taken along each dimension ($4 \times 4 \times 4$ for a 3D grid); so that the point, in which the derivative will be calculated, as far as possible, must be inside the central interval (Fig. 4). This causes a change of interpolation polynomial when

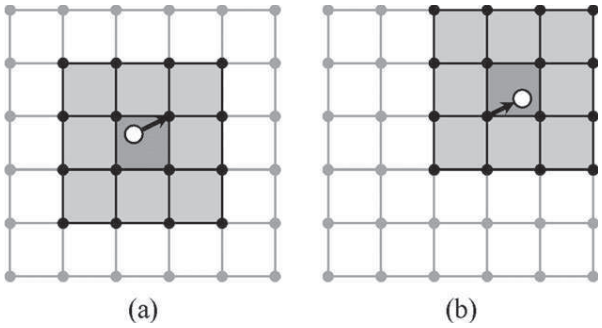

Figure 4. Change of 2D interpolation polynomial. In any point within in the dark gray zone, the values f_{k_1k2k3} used by the interpolation polynomial, eq. (8), correspond to the 16 black dots in the grid. Calculating the gradient or the Hessian matrix in several points starting from the open circle in a) and ending at the open circle in b), the interpolation polynomial is steeply changed when the point reaches the dark gray zone in b).

leaving the central interval. In general, this discontinuity does not represent a serious disadvantage, nevertheless, the Newton–Raphson method can fail when the critical point is exactly on the boundary planes of intervals [with a value of the s_i equal to zero, eqs. (7) and (8)], in which case it is possible that the gradient norm always be greater than the selected convergence criteria.

In the developed library, an option exists that allows continuing with the interpolation polynomial previously used, when the point moves out (not more than 10%) of the interval length in each dimension. This permits finding the critical points that lie in the boundary planes of intervals (in fact, finds a point quite near), without sacrificing the convergence criterion that affects the position of the other critical points.

Library Description

The grid is stored in a structure (of C language), denominated _GRD. Functions that read grid files and return a pointer to a _GRD structure with all the information of the grid (the necessary memory space for this structure is dynamically allocated in the functions) are available. The lagrange3D4grd.c file

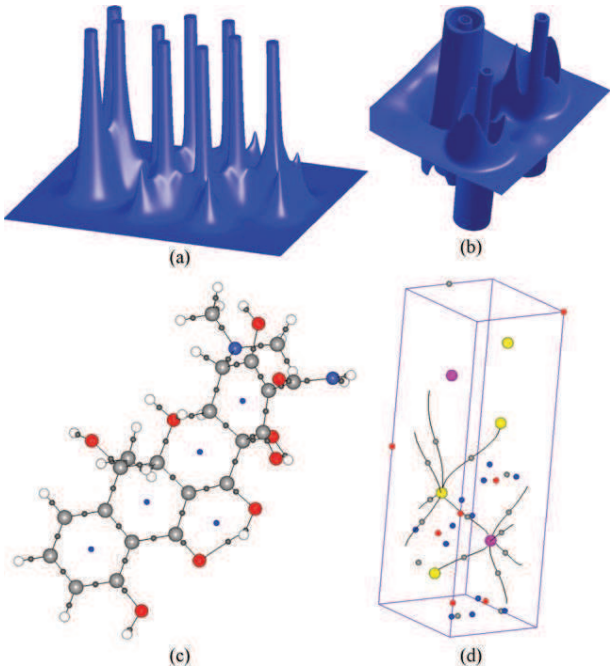

Figure 5. Library application examples: a) Image of the electronic charge density in the plane containing all atoms of a p-nitroaniline molecule. b) Image of the minus Laplacian of the electronic charge density in the plane containing one molybdenum atom and two sulfur atoms of the MoS_2 crystal. c) Bond paths (black lines), and critical points (small gray spheres: bond critical points; blue: ring critical points) of the tetracycline molecule. d) Some bond paths and critical points of the MoS_2 crystal.

contains the read_grd function that reads DMol3 grd files;[35] further description for this function and the _GRD structure are in lagrange3D4grd.h file. GaussianCube.c file contains functions to read grid data in CUBE format from Gaussian[36] and Gamess[37] output files and CHGCARfile.c file the function to read a grid data from Vasp[38] output files.

The C function argument is a character string (**char** * filename) that must contain the name of a *.grd file.

When the _grd_latency global variable has values different from zero it prevents the change of the interpolated polynomial at the interval edges, according to the explanation in Derivative Discontinuities Section.

Functions to calculate the interpolated value, the gradient and Hessian matrix at any space point are described in lagrange3D4grd.h file; also functions to find critical point based on the Newton–Raphson method (described in the Newton–Raphson method Section) and to calculate the points at the gradient path according to the fifth-order Cash–Karp Runge–Kutta method (described in the Fifth-order Cash–Karp Runge–Kutta Method Section).

The library code files can be obtained from the Website: http://alfa.facyt.uc.edu.ve/quimicomp/

Some Examples

The examples shown in Figure 5 are screenshots of graphic windows of some programs that use the described library. The programs were built for Windows XP with MinGW (MinGW is a collection of freely available programming tools, specific header files and import libraries that allow one to produce native Windows® programs Website: http://www.mingw.org/) C compiler and the Dev-C++ (Dev-C++ is a development environment for the C/C++ programming language (for Windows). Free software including a MinGW version Website: http://www.bloodshed.net/devcpp.html) development environment, using OpenGL, Glut, and Glui libraries (Glut and Glui libraries to install it in Dev-C++ can be obtained from the Website: http://www.nigels.com/glt/devpak/).

As an interpolation example, Figure 5a shows the electronic charge density plot on the plane containing all atoms of p-nitroaniline. The calculation was performed using the software Gamess-US[37] at the HF/6-31G(2p,2d) level and the generated grid spacing was 0.1 Bohr. As an example of derivatives calculation with the interpolation of the logarithm, Figure 5b is the graph of the Laplacian of electronic charge density of crystalline MoS_2 on a plane containing one Mo and two S atoms, calculated by density functional theory (DFT) with Perdew-Burke-Ernzerhof (PBE) exchange-correlation functional, and double numerical plus polarization (DNP) basis set using the software DMol^3.[35] Figure 5c is an example of critical points calculation and gradient path for tetracycline (Newton–Raphson and Cash–Karp Runge–Kutta methods, respectively) calculated using the software Gamess-US[37] at the HF/6-31G(2p,2d) level; the grid spacing was 0.05 bohr. In Figure 5d, some bond paths and critical points of the MoS_2 crystal (inclined unit cell) are shown.

For tetracycline, 65 critical points (different from nuclear positions) were determined with a program using this library and also with an analytical electron density method (Multiwfn[26]). The maximal distance, between the calculated critical point position with this method and the Multiwfn program, was 0.0013 Bohr, equivalent to 2.6% of the grid spacing; and 58 of the 65 points have a distance less than 0.0005 Bohr. These differences can be diminished by refining the grid.

Acknowledgments

The authors express thanks to Professor Oscar Valbuena for his help in preparing this manuscript.

Keywords: multidimensional interpolation · QTAIM · C language

How to cite this article: D. Vega, Y. Aray, J. Rodríguez, *J. Comput. Chem.* **2012,** *33,* 2526–2531. DOI: 10.1002/jcc.23083

[1] (a) R. F. W. Bader, Atoms in Molecules-a Quantum Theory; Clarendon Press: Oxford, U.K., **1990**; (b) R. F. W. Bader, *Chem. Rev.* **1991**, *91,* 893.
[2] R. F. W. Bader, *J. Phys. Chem. A* **1998**, *102,* 7314.
[3] R. F. W. Bader, P. L. A. Popelier, T. A. Keith, *Angew. Chem. Int. Ed. Engl.* **1994**, *33,* 620.
[4] P. F. Zou, R. F. W. Bader, *Acta Crystallogr.* **1994**, *A50,* 714.
[5] M. E. Eberhart, *Can. J. Chem. Soc.* **1996**, *74,* 1229.
[6] M. A. Pendás, A. Costales, V. Luaña, *Phys. Rev. B* **1997**, *55,* 4275.
[7] P. L. A. Popelier, Atoms in Molecules—An Introduction; Prentice Hall: Harlow-England, **2000**.
[8] R. J. Gillespie, P. L. A. Popelier, Chemical Bonding and Molecular Geometry: From Lewis to Electrón Densities; Oxford University Press: New York, Oxford, **2001**.

[9] Y. V. Ivanov, E. L. Belokoneva, J. Protas, N. K. Hansen, V. G. Tirelson, *Acta Crystallogr. B* **1998**, *54,* 774.
[10] T. S. Koritsanszky, P. Coppens, *Chem. Rev.* **2001**, *101,* 1583.
[11] A. Fischer, D. Tiana, W. Scherer, K. Batke, G. Eickerling, H. Svendsen, N. Bindzus, B. B. Iversen. *J. Phys. Chem. A,* **2011**, *115,* 13061.
[12] L. J. Farrugia, P. Macchi. *Struct. Bond* **2012**, 1, *146,* 127.
[13] R. Boese, N. Niederprüm, D. Bläser, A. Maulitz, M. Y. Antipin, P. R. Mallinson, *J. Phys. Chem. B* **1997**, *101,* 5794.
[14] P. Munshi, T. N. Guru Row, *Crystallogr. Rev.* **2005**, *11,* 199.
[15] C. J. Mei, K. E. Edgecombe, V. H. Smith, A. Heilingbrunner, *Int. J. Quantum Chem.* **1993**, *48,* 287.
[16] G. H. Grosch, K. J. Range, *J. Alloys Compd.* **1996**, *233,* 39.
[17] M. Knecht, H. Ebert, W. Bensch, *J. Alloys Compd.* **1997**, *246,* 166.
[18] C. Gatti, V. R. Saunders, C. Roetti, *J. Chem. Phys.* **1994**, *101,* 10686.
[19] J. A. Platts, S. T. Howard, *J. Chem. Phys.* **1996**, *105,* 4668.
[20] (a) P. L. A. Popelier, *Comp. Phys. Comm.* **1996**, *93,* 212; (b) P. L. A. Popelier, *Comp. Phys. Comm.* **1998**, *108,* 180.
[21] C. Gatti, An Electron Density Topological Program for Systems Periodic in N ($N =$; 0–3) Dimensions; CNR-ISTM: Milano, **1999**.
[22] AIM2000: F. W. Biegler-König, J. Schönbohm, D. Bayles, Website: http://gauss.fh-bielefeld.de/aim2000. Accessed on February 15, 2012.
[23] M. Barzaghi:PAMoC (Version 2002.0), Online User's Manual; CNR-ISTM, Institute of Molecular Science and Technologies: Milano, Italy, 2002. Website: http://www.istm.cnr.it/∼barz/pamoc/. Accessed on February 15, 2012.
[24] AIMAll (Version 12.06.03), Todd A. Keith, TK Gristmill Software, Overland Park KS, USA, 2012, Website: http://aim.tkgristmill.com. Accessed on February 15, 2012.
[25] Aimpac: http://www.chemistry.mcmaster.ca/aimpac/imagemap/imagemap.htm. Accessed on February 15, 2012.
[26] Multiwfn: T. Lu, F. Chen, *J. Comp. Chem.* **2012**, *33,* 580.
[27] M. Kohout, DGrid 4.6, User's Guide, Max Planck Institute for Chemical Physics of Solids, Dresden, Germany, 2011. Website: http://www.cpfs.mpg.de/∼kohout/Documents/DGrid-4.6.pdf. Accessed on February 15, 2012.
[28] Critic:A. Otero-de-la-Roza, M. A. Blanco, A. Martín Pendás, V. Luaña, *Comp. Phys. Comm.* **2009**, *180,* 157.
[29] Y. Aray, J. Rodriguez, D. Vega, In The Quantum Theory of Atoms in Molecules- From Solid State to DNA and Drug Design; C. Matta, R. Boyd, Eds. Wiley-VCH, **2007**; pp 231–256.
[30] Y. Aray, J. Rodríguez, D. Vega, S. Coll, E. Rodríguez-Arias, F. Rosillo, *J. Phys. Chem. B* **2002**, *106,* 13242.; (b) Y. Aray, M. Marquez, J. Rodríguez, S. Coll, Y. Simón-Manso, C. Gonzalez, D. Weitz, *J. Phys. Chem. B* **2003**, *107,* 8946.; (c) Y. Aray, M. Marquez, J. Rodríguez, D. Vega, Y. Simón--Manso, S. Coll, C. Gonzalez, D. Weitz, *J. Phys. Chem. B* **2004**, *108,* 2418.; (d) Y. Aray, J. Rodríguez, S. Coll, E. Rodríguez-Arias, D. Vega, *J. Phys. Chem. B* **2005**, *109,* 23564.; (e) Y. Aray, A. Vidal, J. Rodriguez, M. Grillo, D. Vega, D. Coll *J. Phys. Chem. C* **2009**, *113,* 19545.
[31] D. Kincaid, W. Cheney, Numerical Analysis; Brooks/Cole Publishing Company, California, **1991**.
[32] C. Katan, P. Rabiller, C. Lecomte, M. Guezo, V. Oisona, M. Souhassoub, *J. Appl. Cryst.* **2003**, *36,* 65.
[33] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C; Cambridge University Press, New York, **1992**.
[34] D. Vega, Modelaje de la adsorción de moléculas pequeñas sobre superficies metálicas, Tesis Doctoral; Instituto Venezolano de Investigaciones Científicas: Caracas, Venezuela, **2004**; Apéndice B.
[35] DMol^3, User Guide, Release 3.2; Accelrys Software Inc., **2005**.
[36] GAUSSIAN 09 user's reference, 2011: http://www.gaussian.com/g_tech/g_ur/u_cubegen.htm. Accessed on February 15, 2012.
[37] M. S. Gordon, M. W. Schmidt, In Theory and Applications of Computational Chemistry: the first forty years; C. E. Dykstra, G. Frenking, K. S. Kim, G. E. Scuseria, Eds.; Elsevier: Amsterdam, **2005**; pp. 1167–1189.
[38] VASP the GUIDE, Universität Wien, Austria, 2011, Website: http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html.

Received: 17 February 2012
Revised: 12 June 2012
Accepted: 12 July 2012
Published online on 2 August 2012

Journal of Computational Methods in Sciences and Engineering 14 (2014) 131–136
DOI 10.3233/JCM-140491
IOS Press

AIM-UC: An application for QTAIM analysis

D. Vega* and D. Almeida
Laboratorio de Química Computacional, FACYT, Universidad de Carabobo, Bárbula, Edo. Carabobo, Venezuela

Abstract. In this paper we introduce AIM-UC, a free application that allows to create graphs related to Quantum Theory of Atoms in Molecules (QTAIM). The input data are files that contain 3D grid of electron density, in some known formats: CUBE from gaussian, grd from DMol and CHGCAR from VASP. Also it supports wave function files from AIMPAC. The application calculates 2D regular grids (electron densities and its laplacians) in a plane positioned by the user. By using bicubic interpolation and the Newton-Raphson method, the critical points at the plane can be calculated, and the gradient field and molecular graph can also be generated using the Cash-Karp Runge-Kutta method. It also permits plotting the electron density and its negative laplacian on the selected plane. Moreover, for both 2D grids, it is possible to calculate the contour maps using the "Marching Squares" algorithm. This application also permits finding the critical points in the 3D space. The output consist of plain text files and images in PNG, BMP and EPS (encapsulated post script) formats. AIM-UC was programmed in C++, using the GUI toolkit FLTK (fast light toolkit) and OpenGL (open graphic library).

Keywords: QTAIM, free software, 2D grid, 3D grid

1. Introduction

The quantum theory of atoms in molecules (QTAIM) allows to represent chemical bonds, predicting its multiplicity and delimiting the space occupied by each atom in molecules and crystalline solids, through topological analysis of the electron charge density. It is also possible to represent electron concentrations (analogous to unshared pair of Pauling) by analyzing the topology of the laplacian of the electron density. The QTAIM [1,2] was developed by Richard Bader [3] in the 70s, and it has been useful in the analysis of properties of systems with ab-initio [4–6] and experimental [7–9] electron densities.

There are several applications for QTAIM analysis, among them free open source: Aimpac [10], Dgrid [11], Critic [12], PAMoC [13] with no graphical interface and Multiwfn [14] with a limited graphical interface. Others with good graphics interfaces, but are not free: AIMAll [15] and AIM2000 [16]. Let us clarify that the list of cited applications is not an exhaustive one. In our laboratory we developed a free application whit a friendly graphical interface, AIM-UC, which generates some QTAIM related graphics (gradient field, molecular graph and contour map). Moreover it permits plot the electron density and its negative laplacian. Although in the near future there are no plans for releasing the source code, versions are available for some operating systems: Microsoft Windows 32 and 64 bits, and the following 64-bit Linux: Centos 6.3, Ubuntu 12.10 and Xubuntu 12.10.

*Corresponding author: D. Vega, Laboratorio de Química Computacional, FACYT, Universidad de Carabobo, Av. Salvador Allende, Edif. FACYT-Química, Bárbula, Edo. Carabobo, Venezuela. E-mail: dvega@uc.edu.ve.

1472-7978/14/$27.50 © 2014 – IOS Press and the authors. All rights reserved

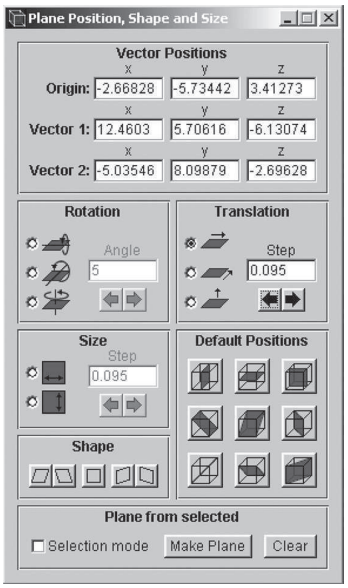

Fig. 1. Application windows containing the tools to position the plane (2D grid).

AIM-UC can read data from files containing electron density grids: Gaussian CUBE [17], DMol GRD [18], VASP CHGCAR [19] among others (see the user manual), and, recently added, AIMPAC wavefunction file [10]. All the displayed images by AIM-UC can be saved in PNG or BMP formats, and the 2D contour map and gradient lines can also be saved in EPS format. The application also allows you to save plain text files with information about the critical points.

A remarkable feature of this application is the set of tools for positioning the plane (2D grid contained in a parallelogram) including displacement and rotation. Also the coordinates of the origin and the two vectors (not necessarily orthogonal) forming the parallelogram can be directly introduced, or the plane can be positioned so that it contains three points (chosen among atomic nuclei and critical points). Figure 1 shows the window that contains these tools.

Other important feature is the ability to work with huge 3D grids. The grid is stored in the memory by dynamic allocation, using a pointer to vector of pointers to vectors of data pointers (a triple pointer), allocating a maximum of continuous memory equal to the data size multiplied by the maximum number of points in one dimension. On a 4 GB RAM PC, the application can work with a grid with 900 $\times$ 900 $\times$ 900 points.

Moreover, it supports periodic systems: critical points are not replicated when are placed on the faces of the periodic cell, and the plane is plotted correctly even if part of this is outside of unit cell.

AIM-UC can be obtained, from the website http://alfa.facyt.uc.edu.ve/quimicomp/.

2. 2D grids construction

The electron density and its laplacian 2D grids are constructed using tricubic Lagrange interpolation [20], the interpolated value (P) is given by Eq. (1).

$$P = \sum_{k_1=0}^{3} \sum_{k_2=0}^{3} \sum_{k_3=0}^{3} w_{k_1} \cdot w_{k_2} \cdot w_{k_3} \cdot f_{k_1 k_2 k_3} \tag{1}$$

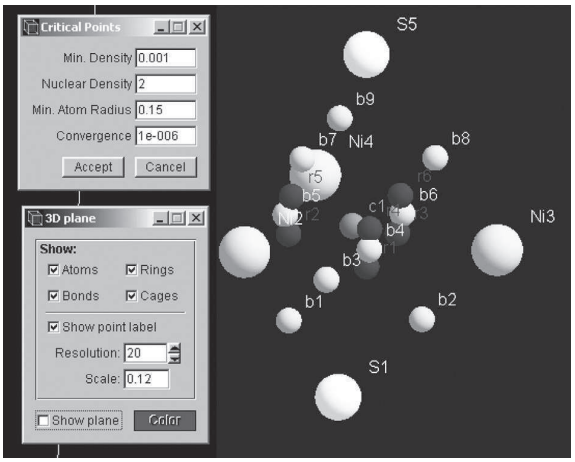

Fig. 2. Critical points found in a Ni_3S_2cluster. Sulfur nuclei are the top and bottom big spheres, nickel are the three big spheres at the middle. There are one cage critical point (labeled as c1), six ring points (dark little spheres) up and down of each Ni-Ni bond critical point, and nine bond points (labeled from b1 to b9).

where w_{k_i} are the Lagrange basis polynomial (depending on the coordinates of the point to be evaluated) and $f_{k_1k_2k_3}$ are the values of the grid points surrounding the point to be evaluated.

When the input data come from an AIMPAC wavefunction file [10], both grids are calculated analytically. The grid is within a parallelogram defined by the origin of coordinates ($\boldsymbol{o}$) and two linearly independent vectors ($\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, not necessarily orthogonal. The density and its laplacian are evaluated at the point (p) coordinates given by Eq. (2).

$$p = o + (i/N_1)v_1 + (j/N_2)v_2 \tag{2}$$

where N_1 and N_2 are the interval numbers in which the vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are divided, and i and j indices run from 0 to N_1 and 0 to N_2, respectively.

If the electron density value is greater than "Nuclear density", a tricubic interpolation of density logarithms [20] is used. Thus preventing that the interpolation polynomial oscillates close to nuclei, where the electron density increases appreciably. An adequate value for "Nuclear density" is half the maximum density of hydrogen atom. This value has no effect in AIMPAC wavefunction files.

3. Critical points

The application allows to find the critical points in space and in 2D grid, using the Newton-Raphson method [21], except near the nucleus positions (maxima), where this method fails due to the high value of the norm of electron density gradient. In order to find the maxima, the algorithm looks for the grid points having density greater or equal than all its nearest neighbors (4 neighbors in 2D and 6 in 3D), then the point with greater density value in the direction of the neighbors, but at half of the distance, is selected. The maxima are the points obtained when this process is repeated several times, always reducing the distance, moving in the same initial directions and using the interpolation of density logarithms. In the case of an AIMPAC wavefunction file, the maximum 3D critical points (near the nuclei) are just the coordinates of the nuclei.

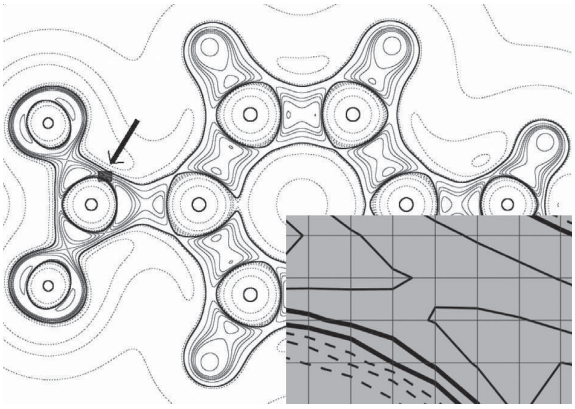

Fig. 3. Contour map of p-nitroaniline molecule. The heavy, the dotted and the solid thin lines correspond to zero, positive and negative isovalues, respectively. The light gray rectangle at the right bottom, is an enlargement to differentiate the grid points (joined by gray lines).

For the remaining critical points, due to the high speed of the interpolation method calculation, a systematic search is utilized. Some points on the grid are the initial points for the Newton-Raphson method and a limited displacement is allowed. In the case of an AIMPAC wavefunction file, the method uses as initial points the average coordinates of each pair, trio and quartet of atoms. Figure 2 shows the critical points for the Ni_3S_2 cluster, found in a density grid with $163 \times 169 \times 188$ points, calculated at the DFT-PBE/DNP level of theory (full geometry optimization) by using the software DMol^3 [18].

4. Gradient fields and isolines

The gradient paths of 2D grids are constructed using the method of Cash-Karp Runge-Kutta [21]. The lines that represent the atomic basins start at the points located on a circle around each maximum, following the negative gradient. The molecular graph is obtained from two paths starting on both sides of each saddle critical point in the direction of the hessian matrix eigenvector (evaluated at the saddle point) with positive eigenvalue, following the gradient. The lines delimiting the atomic basins are also obtained from the saddle points, but in the direction of the other eigenvector, and following the negative gradient.

Not all maxima on the plane correspond to atoms, neither all saddle points corresponding to bond critical points. The application calculates the distance between the plane and the space critical points, hiding the bond and gradient basin paths when there are no coincidence of critical points. The application also allows the user to hide or show, change color, type and width of the lines.

The isolines of the electron charge density and its laplacian are built with the "Marching Squares" algorithm [22]. For each grid square the vertex values and the isovalue are compared. In case of find edges with a vertex with higher value and the other with lower value than the isovalue, the intersection points are calculated by linear interpolation, and these are joined to form the corresponding isoline. The contour map of the laplacian of electron density is shown in Fig. 3, for a 2D grid with 600×400 points containing all the atom positions of a p-nitroaniline molecule. The gray rectangle, pointed by an arrow, is amplified at the right bottom of image. The lines consist of straight segments. The calculation of initial grid was carried out using the software Gamess-US [23] at the HF/6-31G(2p,2d) level and the grid spacing was 0.1 bohr.

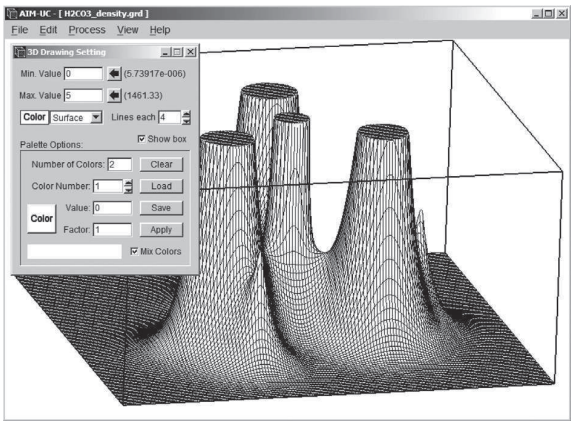

Fig. 4. Electron density plot in a plane containing all atomic positions of carbonic acid molecule. The surface, line and background colors, and the number of lines can be changed.

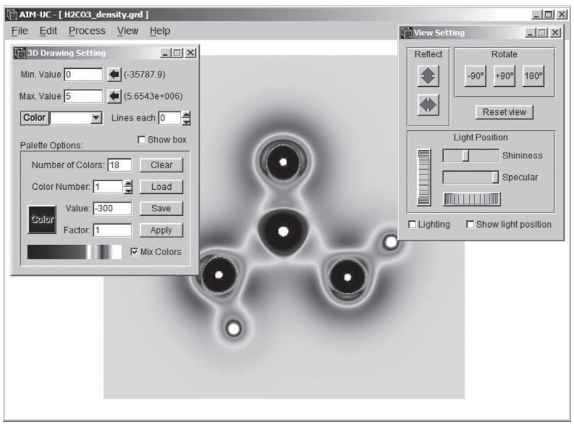

Fig. 5. Laplacian of electron density for carbonic acid molecule, top view, without lighting and colored by value.

5. Final comments

The initial AIM-UC purpose was to perform 2D drawing and further calculating critical points in 3D to determine whether the bond and gradient paths are displayed. However, we are currently working on 3D drawings, adding topology of the electron density (bond paths and interatomic surfaces) and its laplacian (atomic graph), and including the atomic charges calculated by numerical integration.

Except for the rectangle added to Fig. 3, all figures are screenshots of the application, Other AIM-UC capabilities are shown in Figs 4 and 5. The electron density plot of carbonic acid molecule in a plane containing all atoms (Fig. 4), and the negative laplacian of electron density (colored according to their value) of the same molecule on the same plane (Fig. 5), the input grid was calculated at the DFT-PBE/DNP level by using the software DMol^3.

Versions of AIM-UC were compiled under Microsoft Windows 64-bit operating system using TDM-GCC [24], Windows 32-bit with MinGW [25] and Linux with GCC. We also used external libraries FLTK 1.x [26] for building the user interface, OpenGL for graphics and libpng [27] to generate PNG

image files. For further information please read the AIM-UC user manual, which can be obtained from the website mentioned at the end of the introduction section.

Acknowledgment

This work was supported by the grant G2005000424 from the Fondo Nacional de Ciencias y Tecnología (Fonacit) of Venezuela.

References

[1] R.F.W. Bader, *Atoms in Molecules-a Quantum Theory*, Clarendon Press, Oxford, UK, 1990.
[2] R.F.W. Bader, Atoms in molecules-a quantum theory, *Chem Rev* **91** (1991), 893–928.
[3] Obituary in Canadian Chemical News: http://www.accn.ca/index.php?ci_id=3370&la_id=1. Accessed on April 15, 2013.
[4] F. Cortés-Guzmán and R.F.W. Bader, *Coord Chem Rev* **249** (2005), 633–662.
[5] D. Coll, F. Delbecq, Y. Aray and P. Sautet, *Phys Chem Chem Phys* **13** (2011), 1448–1456.
[6] M. Jabłoński and M. Palusiak, *J Phys Chem A* **116** (2012), 2322–2332.
[7] L.J. Farrugia, C. Evans, H.M. Senn, M.M.Hänninen and R. Sillanpää, *Organometallics* **31** (2012), 2559–2570.
[8] P. Sledz, R. Kaminski, M. Chruszcz, M.D. Zimmerman, W. Minor and K. Wozniak, *Acta Cryst* **B66** (2010), 482–492.
[9] V.A. Zhurov, E.A. Zhurova and A.A. Pinkerton, *Inorg Chem* **50** (2011), 6330–6333.
[10] Aimpac: http://www.chemistry.mcmaster.ca/aimpac/imagemap/imagemap.htm. Accessed on April 15, 2013.
[11] M. Kohout, DGrid 4.6, User's Guide, Max Planck Institute for Chemical Physics of Solids, Dresden, Germany, 2011. Website: http://www.cpfs.mpg.de/∼kohout/Documents/DGrid-4.6.pdf. Accessed on April 15, 2013.
[12] A. Otero-de-la-Roza, M.A. Blanco, A. Martín Pendás and V. Luaña, Critic, *Comp Phys Comm* **180** (2009), 157–166.
[13] M. Barzaghi, PAMoC (Version 2002.0), Online User's Manual; CNR-ISTM, Institute of Molecular Science and Technologies: Milano, Italy, 2002. Website: http://www.istm.cnr.it/∼barz/pamoc/. Accessed on April 15, 2013.
[14] T. Lu and F. Chen, Multiwfn, *J Comp Chem* **33** (2012), 580–592.
[15] AIMAll (Version 12.06.03), Todd A. Keith, TK Gristmill Software, Overland Park KS, USA, 2012, Website: http://aim.tkgristmill.com. Accessed on April 15, 2013.
[16] F.W. Biegler-König, J. Schönbohm and D. Bayles, AIM2000, Website: http://gauss.fh-bielefeld.de/aim2000. Accessed on April 15, 2013.
[17] GAUSSIAN 09 user's reference, 2011: http://www.gaussian.com/g_tech/g_ur/u_cubegen.htm. Accessed on April 15, 2013.
[18] DMol3, user guide, release 3.2, Accelrys Software Inc. 2005.
[19] VASP the GUIDE, Universität Wien, Austria, 2011, http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html. Accessed on April 15, 2013.
[20] D. Vega, Y. Aray and J. Rodríguez, *J Comp Chem* **33** (2012), 2526–2531.
[21] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1992.
[22] A description of this algorithm may be found at the website: http://users.polytech.unice.fr/∼lingrand/MarchingCubes/algo.html. Accessed on April 15, 2013.
[23] M.S. Gordon and M.W. Schmidt, Advances in electronic structure theory: GAMESS a decade later, in: *Theory and Applications of Computational Chemistry: The First Forty Years*, C.E. Dykstra, G. Frenking, K.S. Kim and G.E. Scuseria, eds, Elsevier, Amsterdam, 2005, pp. 1167–1189.
[24] TDM-GCC is a compiler suite for Windows basedon GCC and MinGW: http://tdm-gcc.tdragon.net/. Accessed on April 15, 2013.
[25] MinGW is an open source programming tool set for Windows: http://www.mingw.org/.
[26] FLTK is a cross-platform C++ GUI toolkitfor UNIX/Linux (X11), Microsoft Windows, and MacOS X: http://www.fltk.org/. Accessed on April 15, 2013.
[27] Libpng is the official PNG reference library: http://www.libpng.org/pub/png/libpng.html. Accessed on April 15, 2013.